

Az Országos Commodore Egyesület tagjainak

1987/8-9

C újság

PROGRAMFUTAM: A STARTNÁL C 16-OS, PLUSZ/4-ES RAJZOLÓ PROGRAMOK

RAJZOLÓ PROGRAM A PLUSZ/4-ESHEZ

MELYIK CIPŐ AZ IGAZI? MEGMONDJA A PROGRAM

MAJOMFOGÓ A VC 20-HOZ

A KUKKOLÁS FOLYTATÓDIK

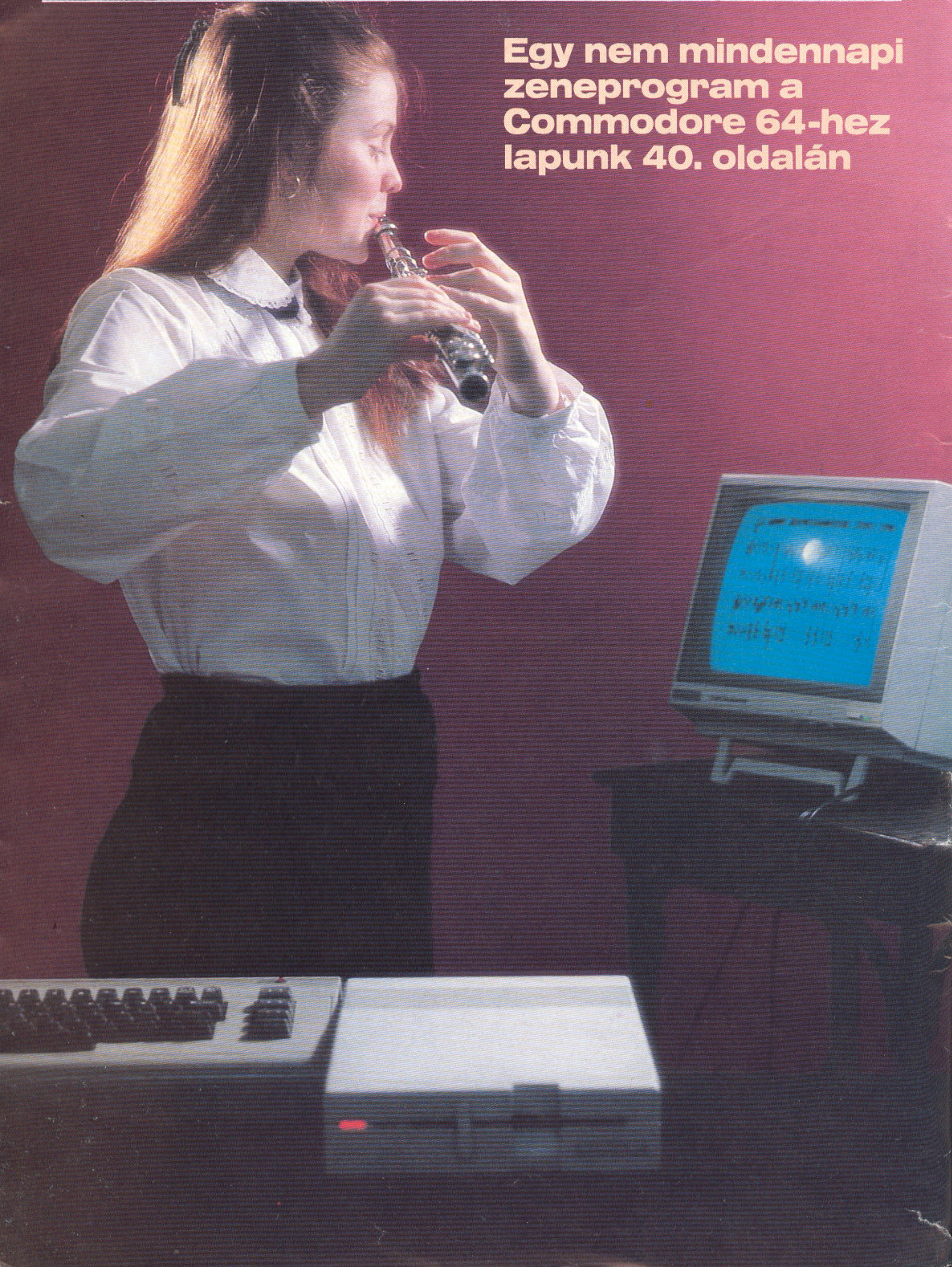
A SCARABEUS SIKERKOVÁCSA

HARDVERESEKNEK: A PLUSZ/4-ES BELSEJE RAJZBAN

COMMODORE KÖNYVLISTA

commo do re

Egy nem mindennapi zeneprogram a Commodore 64-hez lapunk 40. oldalán



Az Országos Commodore Egyesület módszertani kiadványa
Felelős kiadó: Horváth Judit, az egyesület elnöke
Felelős szerkesztő: Angyalosi László
Szerkesztő: Huszerl József, Tallér József
Művészeti szerkesztő: Pribelszky Pál
Tördelő: Domokos Imre
Fotó: Bausz Sándor, Szabó Mihály, Gárdos Katalin, ifj. Novotta Ferenc
Szerkesztőségi titkár: Kollár Gabriella
Levélcím: Commodore Újság Pozsonyi út 50. fsz. 4. 1133
Telefon: 408-603 Index: ISSN 0237-756 X
Készült a Globál GMK gondozásában,
a **Révai Nyomda Egri Gyáregységében**
Felelős vezető: Horváth Józsefné dr. igazgató

A szerkesztő dilemmája

A Commodore Újságnak többféle olvasója van. Van, aki 64-es géppel rendelkezik, van, aki 16-ossal vagy Plusz/4-essel, megint másoknak VC 20-asuk van és végül sokan olvasnak bennünket, akiknek nincs is gépük, vagy már magasabb kategóriákba léptek.

A szerkesztő jól tudja, hogy ki-ki a saját gépére keres programot. Akinek meg nincs gépe, az a programok helyett is szívesebben látna cikkeket, információkat a lapban. Az érdekek tehát ütköznek. Nekünk persze igyekezni kell minden igényt kielégíteni. Tördeljük is minden hónapban a kezünket, hogy mit közöljünk, mit hagyjunk ki, mit tegyünk el a következő hónapra. Döntéseinket természetesen befolyásolja, hogy éppen mink van, milyen programok érkeztek hozzánk az elmúlt hónapokban, de döntéseink hozatalakor persze igyekszünk azt is végiggondolni, hogy hogy elégíthetünk ki lehetőség szerint minden igényt. Mert vannak azután további szempontok is. Ha komolyabb programot tudunk közreadni – az nagy öröm, de több helyet foglal. Azután az is mindig kérdés, hogy egy listát milyen formában jelentessünk meg. Hiszen a legrövidebb forma gyakran a legkevesebbet mondó is. Márpedig a lapunkat olvasók közül sokan csak a programozói tanulságok kedvéért néznek bele egy listába, mások meg tovább szeretnék fejleszteni a mi programunkat, s ehhez kell ismerniük annak lelkét.

Most, hogy dupla számot készítettünk, abban reménykedtünk, hogy mindenkinek a kedvében járhatunk. Minden igényt kielégíthetünk és nem maradnak a lap elkészültével hiányérzeteink. Sajnos nem így történt. Most is kellett döntéseket hoznunk, hogy melyik ujjunkat harapjuk meg. Kimaradt például egy nem mindennapi VC 20-as program, amely 32 új BASIC utasítás használatára ad lehetőséget. De nem fért be az az anyag sem, amely az Amiga 500-as iránt érdeklődőknek adhat új információkat a gépről. Két C 16-os játékprogram futtatását is elvégeztük, de a futam eredményeit is csak az októberi számban adjuk majd közre.

Egyszóval jólesett, hogy 68 oldalban gondolkodhatunk, de mégis maradtak hiányérzeteink. Szerencse, hogy erre aztán igazán érvényes a mondás, hogy ezek örömteli gondok!

ANGYALOSI LÁSZLÓ

AZ IBM® SZTORI

FOLYTATÓDIK!



**Az „óriás hallgat"... írtuk az IBM sztori című cikkünkben idén februárban.
A 32 bites gépek területén mintha lemaradt volna a cég – írtuk –, de a nagy csöndben talán épp arra készült, hogy valami nagy dologgal álljon elő. Nos, érzésünk hamarabb beigazolódott, mint gondoltuk volna. Áprilisban az IBM száz új terméket jelentett be, köztük egy új gépcsaládot, amelynek egyik tagja 32 bites processzort tartalmaz.**

Az új családról azért nem adtunk hírt eddig, mert vártunk. Vártuk, hogy bizonyos, eddig nem ismert fogalmakról kiderüljön valami. Reméltük, hogy a nyugati szaksajtó előbb-utóbb megmagyarázza, hogy mi az az analóg monitor, meg mikrocsatorna stb.
Vártuk, de hiába. Úgy tűnik, az IBM ezúttal nem olyan nagyvonalú, mint a PC megjelenésekor. Úgy tűnik, hogy az új gépcsalád belső titkait ezúttal nem fedik fel.
És még valami. Az IBM ezúttal késik. Az áprilisi bejelentést követően a 30-as modell kivételével a többire csak a megrendeléseket veszik föl, de gépet ezidáig senki sem kapott.
Sajnos, még mi sem láttunk közelről egyet sem a szériából.

ÚJ SZABVÁNY

A bejelentett termékek között szerepel egy fekete-fehér és három színes monitor, három mátrixnyomtató, egy termonyomtató és egy optikai tároló rendszer. Az új személyi számítógép család tagjai pedig: az **IBM 30-as, 50-es, 60-as és 80-as.** A négy új modell a csúcstechnológia terméke. Ez azt is jelenti, hogy az IBM búcsút mondott az eddig elfogadott PC standard-nek. Az új modellek szembetűnő eltérést mutatnak a szokásos PC-től, de még ideiglenesen kötődnek az **MS–DOS** operációs rendszerhez és az Intel processzorokhoz. Egyébként praktikusan minden mást megváltoztattak. **Új videoszabvánnyal és monitorral, új rendszerépítéssel, lemezformátummal, bővítőkártya csatlakozókkal dolgoznak.**

Az új IBM PC-k kapcsán valósult meg először az ún. **SAA** (System Application Architecture – alkalmazói rendszer építés) technológia. Az SAA célja az egységes IBM hardver-rendszer megteremtése. Ez azt jelenti, hogy egységes billentyűzet, monitor és csatlakozófelület kialakítására törekszik, mind a személyi számítógépeknél, mind a miniszámítógépeknél (IBM System/3X) és nagyszámítógépeknél (IBM/370). A szabványosítás **kiterjed** a gépek szolgáltatásaira – menütechnika, egér, funkcióbillentyűk stb. – is, és így a felhasználó nemigen tud különbséget tenni, hogy PC-n vagy nagygépen dolgozik-e. Az egységes hardverrendszer további nagy előnye, hogy a programfejlesztések géptől függetlenül is történhetnek, hiszen a szoftver hordozhatósága a különböző számítógépek között biztosított. A programfejlesztőknek sem szükséges a gépek sajátosságait külön-külön megismerni, mert azok kezelés szempontjából azonosak. Az SAA technológia specifikációihoz szorosan kötődik a gépek operációs rendszere is.

Az új modellek mindegyike üzemeltethető az MS–DOS 3.3 operációs rendszerrel. A továbbfejlesztett MS–DOS változat **12 új paranccsal bővült,** melyek főként az adatok gyors be- és kivitelét támogatják. Például a FASTOPEN paranccsal az adatok gyorsabban olvashatók a háttértárból. Az APPEND és CALL parancsok megkönnyítik az alkönyvtárakban tárolt adatok kezelését. A **DOS 3.3** operációs rendszer futtatható a régi IBM PC-ken is. Az új szoftver megjelenését egy újabb BASIC interpreter is követte, mely szintén a 3.3-as verziószámot kapta.

OS–2

A legfrissebb modellek kapcsán azonban már készül a még modernebb operációs rendszer, az ún. OS–2 (operating system–2). Az OS–2 messzemenően kihasználja az Intel 80286-os mikroprocesszor adottságait. Direkt címzéssel **16 Mbyte-os memóriaterületet kezelhet,** szemben a hagyományos DOS 640 kbyte-os határértékével.

Virtuálisan akár egy Gigabyte-os munkaterülettel is dolgozhat. Nem okoz problémát a 32 Mbyte feletti keménylemezes háttértár kezelése sem. Az OS–2 támogatja az ún. multitasking üzemet, melynek segítségével egyidejűleg több program futtatható. Az operációs rendszer az IBM 50-es, 60-as és 80-as gépeken üzemeltethető. Speciális üzemmódban engedélyezi hagyományos DOS programok futtatását is. Az OS–2 egyébként **AT kompatíbilis** és az alapvető funkciókon kívül tartalmazza a grafikus fejlesztőfelületet is. Az operációs rendszer támogatja az IBM ún. mikrocsatornás BUS-rendszerét, valamint a PC-k hálózatba ill. nagyszámítógéphez való kapcsolását. A Microsoft szoftverház az OS–2 szállítását 1988 januárjára ígéri. Az ára **800 DM** körül várható, mintegy nyolcszorosa a MS–DOS 3.3-ének.

KOMPATIBILITÁS

A 87-es modellek sikere jórészt a kompatibilitás függvénye. Mindegyik modellnél az **5.25"-os** hajlékonylemezes meghajtókat **3.5"-osokra** váltották fel. Az eltérő lemezformátum gondot jelent a kompatibilitás szempontjából. A probléma kiküszöbölésére az IBM négyféle megoldást ajánl:

- **Soros vagy párhuzamos porton a programok közvetlenül átjátszhatók.**
- **Mindegyik modellhez külső 3.5"-os meghajtó csatlakoztatható.**
- **Mindegyik modellhez külső 5.25"-os meghajtó csatlakoztatható.**
- **A PC-k hálózatba köthetők.**

A programok vagy adatok átmásolása más lemezformátumra azonban csak nem védett esetekre érvényes.

A DOS 3.3 operációs rendszer alatt nem feltétlenül futtathatók a hagyományos PC programok. Ez természetesen azon programoknál áll fenn, amelyek valamilyen módon használják a DOS 3.3 rendszert. A szoftver kompatibilitás mellett lényeges szempont a **hardver kompatibilitás** is. Ebből a szempontból **csak az IBM 30-as megfelelő.** A 30-as modellhez nem okoz gondot a meglévő bővítőkártyákkal való csatlakozás. A PC sorozat nagyobb tagjai más csatlakozófelülettel rendelkeznek.

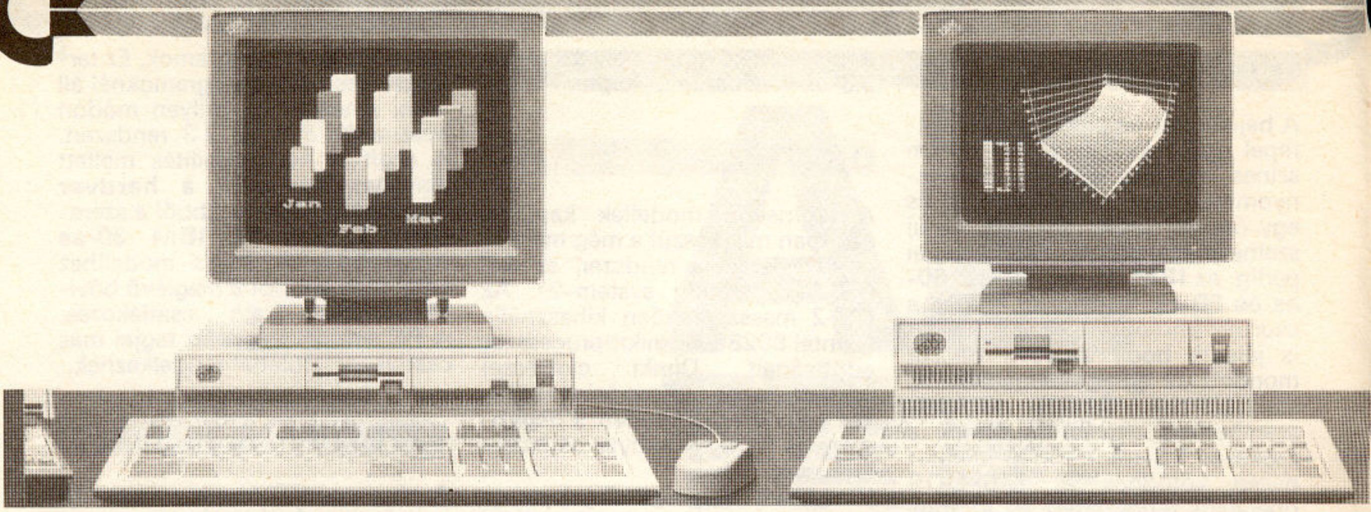
SZÍNORGIA

A négytagú PC-család teljesen **új monitorszabványokat** vezetett be. A számítógépekhez csak az ez évben piacra került IBM **analóg monitorok** csatlakoztathatók. Az IBM 30-as modellnél a rajzolást egy ún. **MCGA** (Multi Color Graphics Array) grafikus adapter támogatja. Az MCGA kártya segítségével a monitoron 320 × 200 pont jeleníthető meg 256 színben egy 266 000-es színpalettából. Monokróm monitorral 640 × 480 pont jelezhető ki 64 árnyalatban. Az IBM 50-es, 60-as és 80-as modellek az MCGA kártya (Video Graphics Array) karaktereket **6 × 16-os** pontmátrixban tárolja és soronként **80 karakter** megjelenítésére képes. A képernyőre egyszerre **43 sor** írható fel. A 30-as modellnél vázolt kijelzési adatok mellett a VCG kártya további szolgáltatása még, hogy **640 × 480 pontot 16 színben** képes megjeleníteni. Készül egy **8514/A adapterkártya, mellyel egy 16"-os monitoron 1024 × 768 pont jeleníthető meg 256 színben egy 262 000-es színpalettából.** Ezen grafikai kártyán mintegy 1 Mbyte-os memóriaterület programozható.

TITKOLÓDZÁS

Az OS–2 operációs rendszernél már jeleztük, hogy az új modellek ún. mikrocsatornás BUS-rendszerrel (Micro-Channel-System) rendelkeznek.

A belső 32 bites sínrendszer párhuzamos adatforgalmát egy DMA (Direct Memory Access – közvetlen tárhozzáférésű) chip vezérli. A vezérlőjelek számára 15 csatorna biztosított, szemben a PC-knél szokásos 2 vonallal. Az IBM féltve őrzi ezen rendszer dokumentációit, hogy megakadályozza az új gépek másolását. A beépítésre került berendezésorientált integrált áramkörök alkalmazása szintén az óriás piaci érdekeit óvja.



IBM 30

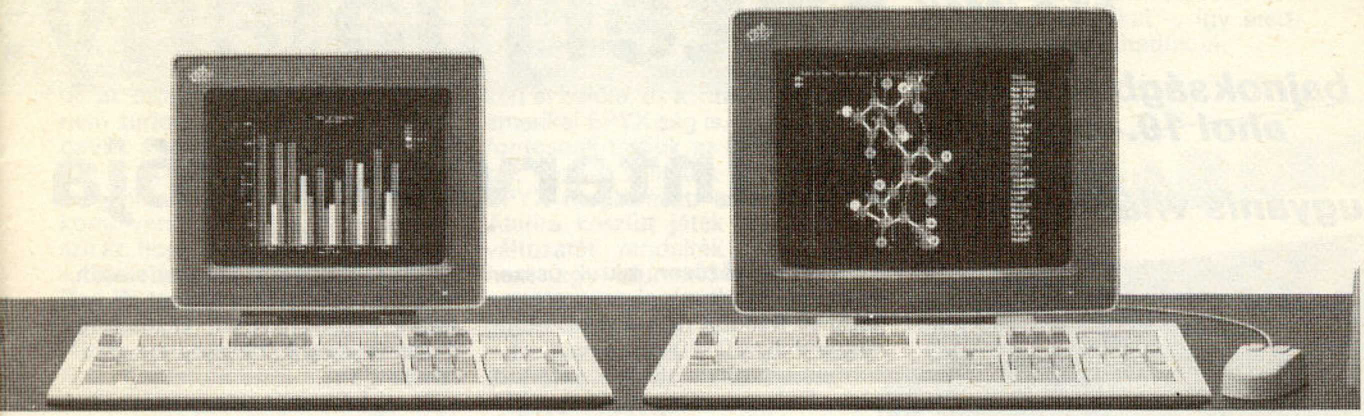
A PC sorozat legkisebb tagja az IBM 30-as. Ez a modell még követi az elődöket, de képességeiben fölülmúlja azokat. A gép a 16 bites Intel 8086-os mikroprocesszorral dolgozik, de kétszer gyorsabb mint a hagyományos XT modellek. Alapkiépítésben 640 kbyte-os RAM-területtel, két 3.5"-os hajlékonylemezes meghajtóval és monokróm monitorral (fekete-fehér) 4400 DM-ért kapható. Az egyik lemezmeghajtó kicserélhető egy 20 Mbyte-os winchesterre, mely 1300 DM-ás árnövekedést von maga után. (Az IBM kedvező árajánlata komoly kihatással van a klóngyártók piaci részesedésére és így joggal nevezik a 30-as modellt a „klón-gyilkosnak".)

IBM 50

Az új gépek alapmodellje az IBM 50-es Intel 80286-os processzorral üzemel, és elsőként rendelkezik mikrocsatornás BUS-rendszerrel. Az egy Mbyte-os munkaterülete 7 Mbyte-ig bővíthető. Háttértárként egy 3.5"-os hajlékonylemezes és egy 20 Mbyte-os keménylemezes meghajtóval dolgozik. Az 50-es modellre a DOS 3.3 mellett majd futtatható a legmodernebb OS–2 operációs rendszer. Az IBM 50-es kétszer gyorsabb mint az XT 286-os, de nem kompatíbilis. Egy alapkonfiguráció kb. 8700 DM-ért kapható.

MŰSZAKI JELLEMZŐK

Modell: Típusszám	30 8530–002	30 8530–021	50 8550–021	60 8560–041
Processzor:	8086	8086	80286	80286
Órajel:	8 MHz	8 MHz	10 MHz	10 MHz
Cooprocesszor:	8087–2	8087–2	80287	80287
Operációs rendszer:				
DOS 3.3:	igen	igen	igen	igen
OS–2:	–	–	igen	igen
ROM:	64 Kbyte	64 Kbyte	128 Kbyte	128 Kbyte
CBIOS (PC/AT kompatíbilis):	–	–	igen	igen
ABIOS (OS–2 kompatíbilis):	–	–	igen	igen
ROM–BASIC:	igen	igen	igen	igen
RAM:	640 Kbyte	640 Kbyte	1 Mbyte	1 Mbyte
Bővíthető max.:	2,6 Mbyte	2,6 Mbyte	7 Mbyte	15 Mbyte
Elérési idő:	150 ns	150 ns	150 ns	80 ns
Adatbusz:	16 bit	16 bit	16 bit	16 bit
Címbusz:	16 bit	16 bit	24 bit	24 bit
DMA–csatornák száma:	3	3	15	15
Háttértárak:				
3,5"-os meghajtó:	2*720 Kbyte	720 Kbyte	1,44MKbyte	1,44 Mbyte
Winchester:	–	20 Mbyte	20 Mbyte	44 Mbyte
Közepes elérési idő:	–	80 ms	80 ms	40 ms
Második winchester opció:	–	–	–	–
Bővítő port:				
PC–kompatíbilis (8 bit):	3	3	–	–
IBM mikrocsatornás BUS:	–	–	3	7
– 16 bites:	–	–	3	7
– 32 bites:	–	–	–	–
Lemezmeghajtó:	–	–	1	1
Winchester:	–	–	–	–
Analóg videóadapter:				
MCGA				
640 * 480, 2 szín:	igen	igen	igen	igen
VGA				
640 * 480, 16 szín	–	–	igen	igen
8514/A opció:	–	–	igen	igen
Interface:				
Soros:	1	1	1	1
Adatátviteli sebesség (baud):	9 600	9 600	19 200	19 200
párhuzamos:	1	1	1	1
Multifunkcionális billentyűzet és „egér":	1+1	1+1	1+1	1+1
Ár (DM):	**4 400**	**5 700**	**8 650**	**15 485**



IBM 60

A 60-as modell részben megfelel a mai AT gépeknek, de pl. kétszer gyorsabb mint az AT03-as. A felhasználó az alkalmazási céltól függően 40 vagy 70 Mbyte-os winchesterek közül válogathat, de szükség esetén második keménylemezes tárat (70 vagy 115 Mbyte) is választhat. A gép meglepő újdonsága, hogy az alapkészülék íróasztal alá került. Körülbelüli ára kiépítéstől függően 15 000 és 17 000 DM között mozog.

IBM 80

A csúcsmodell mindenképpen az IBM 80-as, amely az első 32 bites IBM személyi számítógép. Az Intel 80386-os processzor 16 vagy 20 MHz-es frekvenciákon dolgozhat. Az IBM 80-as 20-szor gyorsabb, mint az első IBM PC, és 3-szor sebesebb mint az AT03-as. A gép alapkiépítésben 1 vagy 2 Mbyte-os munkaterülettel rendelkezik, és ez akár 22 Mbyte-ig is bővíthető. A keménylemezes tárak lehetnek 44, 70 vagy 115 Mbyte-os kapacitásúak és max. 230 Mbyte-ig kiépíthetők. Az ez év végén szállítandó egyes verziók árai 17 000 és 21 000 DM körül várhatók.

60 8560–071	80 8580–041	80 8580–071	80 8580–111	Modell Típusszám
80286	80386	80386	80386	Processzor
10 MHz	16 MHz	16 MHz	16 MHz	Órajel
80287	80387	80387	80387	Cooprocesszor
				Operációs rendszer
igen	igen	igen	igen	DOS 3.3
igen	igen	igen	igen	OS–2
128 Kbyte	128 Kbyte	128 Kbyte	128 Kbyte	ROM
igen	igen	igen	igen	CBIOS (PC/AT kompatíbilis)
igen	igen	igen	igen	ABIOS (OS–2 kompatíbilis)
igen	igen	igen	igen	ROM–BASIC
1 Mbyte	1 Mbyte	2 Mbyte	2 Mbyte	RAM
15 Mbyte	20 Mbyte	20 Mbyte	22 Mbyte	Bővíthető max.
80 ns	80 ns	80 ns	80 ns	Elérési idő
16 bit	32 bit	32 bit	32 bit	Adatbusz
24 bit	32 bit	32 bit	32 bit	Címbusz
15	15	15	15	DMA–csatornák száma
				Háttértárak
1,44 Mbyte	1,44 Mbyte	1,44 Mbyte	1,44 Mbyte	3,5"-os meghajtó
70 Mbyte	44 Mbyte	70 Mbyte	115 Mbyte	Winchester
30 ms	40 ms	30 ms	28 ms	Közepes elérési idő
70/115 Mbyte	44 Mbyte	70/115 Mbyte	115 Mbyte	Második winchester opció
				Bővítő port
–	–	–	–	PC–kompatíbilis (8 bit)
7	7	7	7	IBM mikrocsatornás BUS
7	4	4	4	– 16 bites
–	3	3	3	– 32 bites
1	1	1	1	Lemezmeghajtó
1	1	1	1	Winchester
				Analóg videóadapter
				MCGA
igen	igen	igen	igen	640 * 480, 2, szín:
				VGA
igen	igen	igen	igen	640 * 480, 16 szín
igen	igen	igen	igen	8514/A opció
				Interface
1	1	1	1	Soros
19 200	19 200	19 200	19 200	Adatátviteli sebesség (baud)
1	1	1	1	párhuzamos
1+1	1+1	1+1	1+1	Multifunkcionális billentyűzet és „egér"
16 193	16 890	20 332	21 150	Ár (DM)

Mottó:
Ez a játék egy olyan bajnokságban folyik, ahol 19.-nek lenni is valami. Ez ugyanis világverseny.

Interjúalany: az Interview írója



ADATLAP

Név: CSERI ISTVÁN

Életkor: 26 ÉV

Családi állapot: NŐS, 1 GYERM.

Képzettség: ÉPÍTŐMÉRNÖK

Beosztás: SZELL. SZABADFOGL.

Előélet:
(programokban kifejezve)
INTERVIEW, SAKK, SCARABEUS, INTERNATIONAL KARATE, TURBO ESPIRIT, IMPOSSIBLE II.

– Mondja el, kérem, az Interview történetét!

– Egy munkanélküli táviratot kap, amelyben közlik vele, hogy ha tizenöt percen belül jelentkezik felvételi beszélgetésre – amit angolul interjúnak neveznek –, akkor álláshoz jut. Útját különböző akadályok nehezítik. Például ha banánhéjra lép és elesik, akkor a ruháját ki kell tisztíttatnia, ha pedig kutya harapja meg, akkor az elsősegélyhelyet is útba kell ejtenie.

– Ez volt az első komolyabb játékprogramja. Netán hasonló élményei voltak, amikor programozói megbízatás után járt?

– Nem, a legelső feladatom a Novotrade-nél szinte magától jött, és azóta is nekik dolgozom.

– Kezdjük akkor kicsit korábbról! Hogyan került kapcsolatba a számítástechnikával? Hiszen végzettsége nem szorosan ilyen irányú.

– Régebben a szomszédomban lakott Barna Péter, akivel azóta is együtt dolgozom. Néhány évvel ezelőtt Kit-ben kapott egy ZX–81-et. A gépet közösen raktuk össze, és együtt kezdtünk el programozgatni is. Később ő javasolta, hogy magamnak VC–20-at vegyek, akkoriban az volt az elérhető legjobb gép. Beleszerettem, és egy-két évig egyedül ismerkedtem a géppel – autodidakta módon tanultam meg programozni.

– Hogy jutott el az első játékprogram megírásáig?

– A Novotrade játékpályázata éppen jókor jelent meg. Akkor már biztosnak éreztem magam a programozásban. Írtam egy játékot és beadtam, de elutasították. Pedig teljesen újszerű volt, rosszul tettem, hogy nem kardoskodtam az igazam mellett. Háromdimenziós volt a játék, akkoriban még nem léteztek ilyenek.

– Úgy látszik, hogy nem törte le a kudarc.

– Nem, sőt ismét a Novotrade-hez fordultam, hogy adjanak programozói munkát. Ekkor kaptam meg feladatul az Interview kidolgozását.

– Az már kiderült, hogy az alapötlet nem öntől származik. Honnan ered?

– Egy angol cég rendelte meg a programot, az ötlet az övék volt. Féléves határidőt adtak a megírásra. Kicsit csúsztam vele, de végül is elkészült a játék. Az ötlet persze tényleg csak az alapelképzelés volt, a forgatókönyvben is sok mindent nekem kellett kidolgozni.

– Ez a játék nem túlságosan ismert itthon. Csak Angliában terjesztik?

– Ott sem. Mire elkészült a program, csődbe jutott a megrendelő, megszűnt a cég. Engem szerencsére kifizettek, de a program azóta is a fiókban lapul. Mindenesetre jó iskola volt.

– Következett a Scarabeus. Ezzel bizonyára több szerencséje volt, hiszen a játék itthon is elterjedt.

– Igen, és itt az ötlet is magyar: egyik munkatársam, Baumann Gábor bátyjától származik. A játék labirintusának az az újszerűsége, hogy tényleg háromdimenziós: a játékos fordulása is jól láthatóan, fázisonként követhető, szinte mint egy rajzfilmen. Az alapötletet a Novotrade dolgozta ki részletesen, majd megrendelést kapott rá az Ariola cégtől. Ezután kezdtük el társaimmal együtt a program elkészítését.

– Ez tehát már csapatmunka volt. Hogyan alakult meg a csoport, és mi a nevük?

– Szinte véletlenszerűen jöttünk össze, nevünk azóta sincs. A Novotrade összehozta a közös munkára azokat az embereket, akiknek éppen nem volt feladatuk. Mindenesetre jól sikerült, mert azóta is együtt dolgozunk a Novotrade „ANDROMEDA" Stúdiójának.

– Ki a vezetőjük, és hogy folyik az együttes munka?

– Vezető nincs köztünk. Arra persze ügyelünk, hogy minden projectnek, minden új programnak legyen egy felelőse. Általában az, aki a program vezérlő részét írja, hiszen neki kell összefognia a társaságot.

– A honoráriumok szétosztásában milyen szerepet játszik a project felelőse?

– Szavazással döntünk arról, hogy ki milyen hányadot kapjon egy-egy program megírása után.

– Térjünk vissza a játékokhoz! Mennyi idő alatt készült el a Scarabeus?

– Erre is fél évet kaptunk, és itt nem is volt csúszásunk – bár erre az időszakra esett a diplomám megszerzése is.

– Lediplomázott, mint építőmérnök. Rögtön ezután szabadfoglalkozású lett, vagy elhelyezkedett valahol?

– Kaptam egy hároméves akadémiai ösztöndíjat az Útépítési Tanszéken, így bent maradtam az egyetemen. Egyébként a diplomamunkám is egy számítógépprogram volt: egy hossz-szelvény optimalizáció, amit azóta is használnak az oktatásban. Akkor még úgy láttam, hogy emellett tudom végezni a programozói munkát is. Egy évet töltöttem ott, és rájöttem, hogy a kettő nem megy együtt. A programozást nem akartam abbahagyni, az egyetemi munkára pedig nem jutott elég időm. Nem szeretek semmit sem félig-meddig csinálni, így hát önállósítottam magam.

– Mi lett a Scarabeus további sorsa?
– Átvette a megrendelő, és Angliában huszonötezer példányban értékesítették. Eladták az USA-ba is, de az ottani értékesítési adatokról nem tudok. Folyamatosan készítgetjük a továbbfejlesztett változatot, a Scarabeus II-t is, de idáig még nem volt annyi időnk, hogy komolyan nekiálljunk. A fő ok persze az, hogy nincs rá szerződésünk.
– Következett a sakkprogram. Ezt is megrendelésre készítették?
– Igen, az amerikai Sierra rendelte Atari gépre, de erről a programról nem szívesen beszélek, nagy kudarcunk volt.
– Hogyhogy?
– Elvállalta valaki, hogy megírja a játék algoritmusát. Megbízhatatlan ember volt, és nem is értett igazán hozzá. Ígérgetett, hitegetett. A megrendelőnk pedig előírta, hogy programunknak meg kell vernie a Sargon III-at – ezt pedig a mai napig nem sikerült teljesíteni. Azóta is próbálkozunk a fejlesztésével, az eladásával, de egyre kisebb eséllyel. Több hónapunk ráment, és az egyetlen haszon annyi volt, hogy megismerkedtünk az Atari géppel.
– A Karaténél már újra sikerről beszélhetünk?
– Igen, bár ez a program két szempontból is eltér például a Scarabeustól. Egyrészt ez ismét önálló munkám volt. Gyakorlatilag egyedül dolgoztam. Másrészt itt nem a nulláról kellett elindulni, mert ez a program már létezett C–64-en, a megrendelő az Atarira átírt változatát kérte.
– Milyen eredményt értek el a programmal?
– Angliában jelentős mennyiséget kelt el belőle, és azóta megvette az amerikai EPYX cég is. Egyébként is fontosnak tartjuk az amerikai piacot, ezért vállaltuk szívesen az EPYX megbízását, amelyben két, Atarira készült játék monochrom változatát rendelték meg. Ezek nem nagy volumenű munkák, a lényeg az, hogy a nevünk jelen legyen az USA-ban is.
– Ha jól tudom, a Turbo Esprit C–64-es változata is átirat.
– Igen, a program megvolt már Spectrumon, ezért eleinte könnyűnek is éreztük a feladatot. Még attól sem riadtunk vissza, hogy az eredeti játékot készítő Durell cég lehetetlennek ítélte az átültetést. Két társam elvállalta, és azután egyre több nehézségbe ütköztek. Nagy gondot jelentett a játék változatos helyszíneinek előállítása és tárolása. Ennyi háromdimenziós kép nem fér egyszerre a memóriába, ezért ezeket a program futása közben kell generálni. Ez a játék egyáltalán nem a C–64-re való, eredetileg is a Z–80-as processzor lehetőségeinek minél teljesebb kiaknázására fejlesztették ki. Becsapós munka volt – és valljuk be, társaimnak sem volt elég tapasztalata hozzá. Végül is – amikor a Novotrade némi nyomást gyakorolt ránk – közösen álltunk neki. Ezután elkészítettük annyi idő alatt, mint amennyi az eredeti határidő lett volna, de az addigra persze már régen lejárt. A megrendelő ennek ellenére átvette – ők tisztában voltak a nehézségekkel –, így eredményként könyvelhetjük el.



MŰHELYTITKOK

Hogyan fogok hozzá egy program megírásához? Ez attól függ, hogy egyedül vágok-e bele, vagy csoportmunkáról van szó. Ha az alapötlet megvan, akkor a legelső tennivaló egy részletes forgatókönyv elkészítése, ami a játék valamennyi lehetőségére kitér, és egy folyamatábra megrajzolása, amely minden programrészlet helyét és feladatát megmutatja. A folyamatábra alapján lehet azután szétbontani a munkát részfeladatokra.
Nem szabad félni a teljesen önállóan végzett munkától: aki tud programozni, bármit meg tud oldani egyedül – legfeljebb hosszabb idő alatt, mint ha társakat keresne. A csapatmunkának két nagy előnye van. Az egyik az, hogy a program gyorsabban készül, hiszen egy csoport minden tagja más és más területhez ért jobban – így, ha megfelelő feladatot kap, mindenki a legjobb teljesítményt nyújtja. A másik előny, hogy egy csoport több programot is készíthet párhuzamosan. Ha valaki a saját szakterületén végzett egy feladattal, akkor elkezdheti a következő program hasonló részeinek kidolgozását, anélkül, hogy megvárná, míg az első program teljesen elkészül.
Mindez persze csak akkor van így, ha a csoport tagjai jól szervezhető, megbízható emberek. Egyébként megette a fene az egészet, és jobb, ha egyedül dolgozik az ember. A feladatok szétosztása sem könnyű ügy, hogy mindenki elégedett legyen. Elvégre ez nem favágás, ahol mindenki ugyanúgy csapkod a fejszéjével – itt rengeteg szubjektív tényezőt is figyelembe kell venni.
Egy-egy játékprogram elkezdésekor általában a grafika a „szűk keresztmetszet", így ezzel kell indulni – illetve legelőször a grafikussal kell megbeszélni a feladatát. A grafika persze valamennyi játékot végigkíséri, összefonódik a program vezérlőrészével. Ezért ha önállóan dolgozunk, akkor jobb is olyan ötletet választani, amihez nem kell túl kifinomult grafikai tervezés.
A programkészítési módszerek különbözőek. Van, aki egy-egy nagyobb részt egyszerre ír be a gépbe, és csak ezután „lövi be", szűri ki a hibákat – amelyek persze minden program fejlesztése közben elkerülhetetlenül jelentkeznek. Én inkább lépésenként nézem meg, hogy amit begépeltem, azt teszi-e amit kell, működik-e. Embere válogatja, hogy ki melyik módszert követi, nem lehet rá receptet adni.
Rengeteget lehet tanulni a mások által írt programok áttanulmányozásából, és abból, hogy „beléjük piszkálunk". Sok trükk elleshető így. Aki úgy dönt, hogy játékprogramok írására adja a fejét, annak persze nemcsak a szoftvert, hanem a hardvert is ismernie kell. Nincs mese: mostanában szinte évente meg kell tanulni egy-egy új gépet. Ez nem is baj, fejlődik az ember – és addig sem unatkozik. Így például korábban IBM-re senki sem akart játékprogramot írni, de mára kiderült, hogy szükség van rá.
Sokan jártak már úgy, hogy egy programon dolgoztak éjjel-nappal, és emiatt megcsömörlöttek az egész munkától. Időnként szükség van egy kis nyugalomra is. Ilyenkor megfogadja az ember, hogy két napig be sem kapcsolja a gépet – azután ezt vagy be tudja tartani, vagy sem.

Cseri István

– Ahhoz képest, hogy nem is olyan régen egy ZX–81 kittel kezdte az ismerkedést a számítástechnikával, sűrűn váltogatja a gépeket: Spectrum, C–64, Atari...

– Az első két gép, amivel találkoztam – a ZX–81 és a VC–20 –, ma már csak nosztalgia. A C–64-et és az Amstrad-et a Scarabeus és a Karate fejlesztéséhez kaptam, a sakkprogram és a Turbo Esprit elkészítéséhez pedig szükség volt az Atari megismerésére. Mostanában pedig már IBM-en végzem a programfejlesztést. Egyébként jelenleg ez a megszokott, általánosan használt négyes: C–64, Amstrad, Atari, IBM.

– Ilyen összehasonlítási alap birtokában mi a véleménye a C–64-ről?

– Erre a gépre idén, sőt jövőre is nyugodtan lehet szoftvert írni, ha jó a program, lesz rá vevő. Nagyon jó játékgép, nem lehet eltemetni.

– Min dolgozik jelenleg?

– Az Impossible II programon, ami a jól ismert játék folytatása lesz. A megrendelő ismét az EPYX cég, ők adták a forgatókönyvet is. C nyelven írom Atarira és IBM-re – ez a nyelv nagyon támogatja a fejlesztést –, de egyes részei már készen vannak C–64-re is.

– Hogyan tud egyszerre három gépen dolgozni?

– Az Atarit és az IBM-et összekötöttem egymással, így az IBM-en kidolgozott programrészleteket közvetlenül át tudom küldeni. A C–64-ben viszont nem fér el egyszerre a C-fordító és a játék, így erre a gépre az assembly változatot is meg kell írnom.

– Mennyiben fog eltérni az új játék az eredetitől?

– A forgatókönyv szerint most nyolc torony áll egy csoportban, és mindegyiket végig kell járni a küldetés teljesítéséhez. Mindegyik toronynak önálló arculata, „egyénisége" lesz. A figura alakja, mozgása változatlan marad, de a helyiségek három dimenzióban, axonometrikusan lesznek láthatóak.

– Befejezésül egy kérdés a jövőre vonatkozóan: elkötelezte magát a játékprogram-készítés mellett, vagy tervezi más jellegű, alkalmazói szoftverek írását is?

– Bármilyen más szoftver készítéséhez jó és kemény iskola a játékprogram – sokaknak bele is tört a bicskája. Nem ragaszkodom a játékokhoz, és már bennem is felvetődött, hogy az új Atarira rengeteg olyan segédeszközt lehetne készíteni, ami megkönnyíti a mérnöki tervezőmunkát. Ahhoz azonban, hogy valaki ilyesmibe belevágjon, biztos megrendelés és jó anyagi háttér kell, mert rendkívül időigényes feladat. Nem állhatok neki most tervezőprogramot írni, mert akkor éhen halok.

Tallér József



RASTER

A program a Scarabeus játék egy szubrutinja – némileg módosítva –, amely megvalósítja egy kép „áttűnését" egy másikba. Jelen esetben a $0400 képernyőről lehet a $0800-ra és vissza kapcsolni. A program maga $C800-on lesz – de természetesen máshova is lehet fordítani.

Az átkapcsolás előtt mindig elmenti a színmemóriát, és a másik képernyőt tölti fel soronként. A két bufferterület $C000 és $C400.

A „sima" átváltás úgy lehetséges, hogy az extended és multicolor mód egyszerre van beállítva, ezt a grafikus chip nem bírja, és a kép feketére vált – akárhol, karakter közepén is. Ebben a fekete sávban azután bármit el tudunk intézni – akár BITMAP, vagy más módra áttérni –, és ezután egyszerűen visszaállítjuk normál állapotba a chipet.

Ha nem folyamatosan, hanem utasítások kiadásával akarjuk váltogatni a képernyőket, akkor ezt a SYS 51200, illetve a SYS 51203 beírásával érhetjük el.

Az itt közölt program a Profi-Assembler '64 program segítségével íródott, de természetesen bármely más assembler fordítóprogrammal is használható.

Miután a második képernyő a $0800, így a következő kis kétsoros BASIC programmal használat előtt feljebb kell helyezni a BASIC terület alját.

```
5 REM BASIC ALJA $0C00-RA
10 POKE12*256,0:POKE44,12:NEW
```

```
1 REM ********************************
2 REM *    C= UJSAG     SORSZAM: 077   *
3 REM *    RASZTER                     *
4 REM *    PROGRAM:     CSERI ISTVAN   *
5 REM ********************************
6 REM
10 SYS9*4096
20 .OPT OO
30 *= $C800
40 JMP CHANGE1
50 JMP CHANGE2
100 TEMP = $FB
110 TEMP1 = $FD
120 ;
2000 CHANGE1 = *     ; KEP VALTAS ( $0400-ROL $0800-RA )
2010 LDA #$C0        ; SZINMEMORIA
2020 STA TEMP1+1     ; MENTES
2030 JSR COPYCOL     ; $C000
2080 LDA #$C0
2090 STA TEMP
2100 STA TEMP1
2110 LDA #$C7        ; VISSZAMENTES
2120 STA TEMP+1      ; $C400-ROL
2130 LDA #$DB        ; ( AZ 1K VEGEROL
2140 STA TEMP1+1     ; KEZDVE VISSZAFELE )
2141 LDA #240        ; KEZDO RASTER
2142 STA RASTER1+1   ; ERTEKEK
2143 LDA #255        ; MINDEN CIKLUSBAN
2144 STA RASTER2+1   ; MAGABAN A
2150 JSR UP          ; PROGRAMBAN
2160 LDA #8          ; ATIRVA, NEM
2170 STA 648         ; VALTOZOBAN
2180 RTS
2190 ;
2385 ;
2390 UP SEI
2400 R2400 LDA $D011     ; KEP
2410 BMI R2400           ; TETEJERE
2420 R2420 LDA $D012     ; VAR
2430 CMP #20             ;
2440 BNE R2420           ;
2450 LDA #$15            ; ELSO KEP ($0400)
2460 STA $D018
2530 ;
2600 R2600 LDA $D012     ; SAV TETEJE
2610 RASTER1 CMP #$F0    ; RASTER1+1 MINDIG
2620 BNE R2600           ; ATIRVA
2630 LDX #8              ;
2640 R2640 DEX           ; IDOZITES
2650 BPL R2640           ;
2655 ;
2660 LDA #$5B            ; BLANK A
2665 STA $D011           ; KEP KOZEPEN !
2666 LDA #$D8            ; (EXTENDED ES
2667 STA $D016           ; MULTICOLOR MOD)
2670 LDA #$25            ; MASIK KEP ($0800)
2675 STA $D018
2740 ;
2800 R2800 LDA $D012     ; SAV ALJA
2810 RASTER2 CMP #$FF    ; RASTER2+1 MINDIG
2820 BNE R2800           ; ATIRVA
2830 LDX #8              ;
2840 R2840 DEX           ; IDOZITES
2850 BPL R2840           ;
2860 ;
2900 LDA #$1B            ; NORMAL KEP
2910 STA $D011           ; VISSZAALLITAS
2911 LDA #$C8
2912 STA $D016
2920 ;
2930 LDA RASTER2+1       ; HA UJABB
2940 AND #7              ; KARAKTERSOR,
2950 CMP #3              ; AKKOR
2960 BNE CONRAS1         ; EGY SOR SZIN
2970 ;                   ; VISSZAMASOLAS
2980 LDY #39
3000 R3000 LDA (TEMP),Y
3005 STA (TEMP1),Y
3010 DEY
3020 BPL R3000
3030 LDA TEMP            ; KOVETKEZO
3040 SEC                 ; SZINSOR
3050 SBC #40             ; SZAMITASA
3060 STA TEMP
3070 LDA TEMP+1
3080 SBC #0
3090 STA TEMP+1
3091 LDA TEMP1
3092 SEC
3093 SBC #40
3094 STA TEMP1
3095 LDA TEMP1+1
3096 SBC #0
3097 STA TEMP1+1
3100 ;
3110 CONRAS1 = *
3120 DEC RASTER1+1       ; EGY PIXELLEL
3130 DEC RASTER2+1       ; FELJEBB
3140 LDA RASTER2+1
3150 CMP #48             ; HA ELERTE
3160 BEQ ENDUP           ; A LATHATO
3170 JMP R2400           ; KEP TETEJET,
3175 ENDUP CLI           ; VEGE
3200 RTS
3210 ;
3500 CHANGE2 = *     ; KEP VALTAS ( $0800-ROL $0400-RA )
3510 LDA #$C4
3520 STA TEMP1+1
3530 JSR COPYCOL
3580 LDA #0
3590 STA TEMP
3600 STA TEMP1
3610 LDA #$C0
3620 STA TEMP+1
3630 LDA #$D8
3640 STA TEMP1+1
3641 LDA #40
3642 STA RASTER3+1
3643 LDA #55
3644 STA RASTER4+1
3650 JSR DOWN
4000 LDA #4
4010 STA 648
4012 LDA #$15
4014 STA $D018
4020 RTS
4030 ;
4190 DOWN SEI
4200 R4200 LDA $D011
4210 BMI R4200
4220 R4220 LDA $D012
4230 CMP #20
4240 BNE R4220
4250 LDA #$15
4260 STA $D018
4330 ;
4400 R4400 LDA $D012
4410 RASTER3 CMP #$F0
4420 BNE R4400
4430 LDX #8
4440 R4440 DEX
4450 BPL R4440
4460 ;
4470 LDA #$5B
4480 STA $D011
4490 LDA #$25
4500 STA $D018
4501 LDA #$D8
4502 STA $D016
4570 ;
4600 R4600 LDA $D012
4610 RASTER4 CMP #$FF
4620 BNE R4600
4630 LDX #8
4640 R4640 DEX
4650 BPL R4640
4660 ;
4670 LDA #$1B
4680 STA $D011
4681 LDA #$C8
4682 STA $D016
4690 ;
4700 LDA RASTER4+1
4710 AND #7
4720 CMP #3
4730 BNE CONRAS2
4740 ;
4800 LDY #39
4810 R4810 LDA (TEMP),Y
4820 STA (TEMP1),Y
4830 DEY
4840 BPL R4810
4850 LDA TEMP
4860 CLC
4870 ADC #40
4880 STA TEMP
4890 LDA TEMP+1
4900 ADC #0
4910 STA TEMP+1
4920 LDA TEMP1
4930 CLC
4940 ADC #40
4950 STA TEMP1
4960 LDA TEMP1+1
4970 ADC #0
4980 STA TEMP1+1
4990 ;
5000 CONRAS2 INC RASTER3+1
5010 INC RASTER4+1
5020 LDA RASTER4+1
5030 CMP #254
5040 BEQ ENDDOWN
5050 JMP R4200
5060 ENDDOWN CLI
5070 RTS
5160 ;
8000 COPYCOL = *     ; SZINMEMORIA
8001 LDA #$D8        ; MASOLAS ( 1K )
8002 STA TEMP+1      ; (TEMP)-ROL
8003 LDA #0          ; (TEMP1)-RE
8004 STA TEMP
8005 STA TEMP1
8010 LDX #4
8020 LDY #0
8030 COPY LDA (TEMP),Y
8040 STA (TEMP1),Y
8050 INY
8060 BNE COPY
8061 INC TEMP+1
8062 INC TEMP1+1
8070 DEX
8080 BNE COPY
8090 RTS
8100 ;
```

BODY JÁTÉKOK



Kezdetben voltak az akciójátékok. A piac lassan telítődött a lövöldözős, ugrálós programokkal.
A szoftverházak új irányokat kerestek. Ekkor születtek a sportjátékok – amelyek már lehetőséget adtak a számítógép társasjátékként való használatára, bajnokságok kiírására. Valóságos olimpiák megrendezésére adnak lehetőséget ezek a programok – a legkülönbözőbb komoly és tréfás sportágakkal. A sportjátékokkal szinte egy időben jelentek meg az úgynevezett adventurök – kalandjátékok – nem mindennapi üzleti sikert hozva a fejlesztőknek. Előbb pusztán szöveges, később gyönyörű grafikákkal felékesített kalandjátékok voltak. Nem sokkal ezután pedig megszületett az akció-kalandjáték, amely ötvözi a kétféle típus lehetőségeit. Mi jöhet még? – kérdeztük. Nos, a programfejlesztők fantáziája kifogyhatatlan. Úgy tűnik, az újabb üzleti siker felé vezető úton most azok a játékok következnek, amelyekhez nem elég egy kazetta, vagy egy lemez, hanem speciális hardver eszközök is szükségesek!

A Bodylog cég abból indult ki, hogy a gép előtt ülők nyilván szívesen használnák a gépet a sok üléstől amúgy is károsodó egészségük védelmére.
Három különálló rendszert mutattak be a közelmúltban.

EDZŐPROGRAM

Köztudott dolog, hogy az ember a különböző terhelések és szituációk hatására eltérő szívritmussal és testhőmérséklettel reagál. „Melegebb" helyzetekben erősebben ver a szívünk, vagy stresszhelyzetekben (félelem, nyomás stb.) esetleg egyszerre kihűl a kezünk. Ha megfelelő szenzorokkal ezeket az értékeket felfogjuk, az így nyert adatokat egy megfelelően programozott számítógéppel akár mozgássá is alakíthatjuk.
Kapunk tehát egy modult, amelyet a gép bővítő portjába kell csatlakoztatni, és egy hosszabbítóval ellátott hőmérsékletmérőt. Ezen kívül egy kézikönyvet is mellékel-

nek a szerzők, na meg egy zenés kazettát is.
A kézikönyv arra szolgál, hogy bevezessen minket a koncentráció gyakorlásának különböző módszereibe. Elmagyarázzák, hogyan is alakul ki a stressz, hogyan reagál erre a test – például hőmérsékletváltozásokkal. Megtudjuk azt is, miért fontos ezekre a jelzésekre reagálni, és hogyan is kell bánni velük. A könyvben az érdeklődők néhány lazító és a koncentrációt növelő gyakorlatot is találnak, amelyeket könnyű elsajátítani, és érdemes is megcsinálni.
A könyv tanulmányozása után elkezdhetjük a tulajdonképpeni kezelési leírást olvasni, illetve összeszerelhetjük a szükséges hardveres konfigurációt.
A számítógép bekapcsolása után néhány pillanat múlva a modul a „Peace of Mind" játékkal jelentkezik. A játék célja, hogy a főszereplő egy bizonyos távolságról megközelítsen egy kastélyt, és megtekintse az abban található kincseket.
A vezérlés módja azonban meglehetősen szokatlan. **A figurát** se nem a joystick vad tekergetésével, se nem egy paddlest csavargatásával, hanem a saját **testünk reakcióival kell vezérelni.** Ehhez a mellékelt hőérzékelőt rögzítenünk kell egy alkalmas (és szintén mellékelt) eszközzel az egyik ujjunkon. A program az ott mérhető hőmérsékletet állandóan regisztrálja. Minél nyugodtabb a játékos, annál magasabb a bőrének hőmérséklete, s a játékfigura is annál gyorsabban mozog. Ha azonban idegesek vagyunk, vagy stresszes állapotba kerültünk, akkor a hőmérsékletünk is csökken, s ezzel a sebességünk is kisebb lesz. Extrém esetben a mért érték olyannyira alacsony lehet, hogy a figura nem a kastély felé, hanem az ellenkező irányba kezd el mozogni. A játék célja tehát, hogy a nyugalom kialakításával bőrünk hőmérsékletét stabilizáljuk, hogy a figura a kastély felé haladjon.

FITNESS

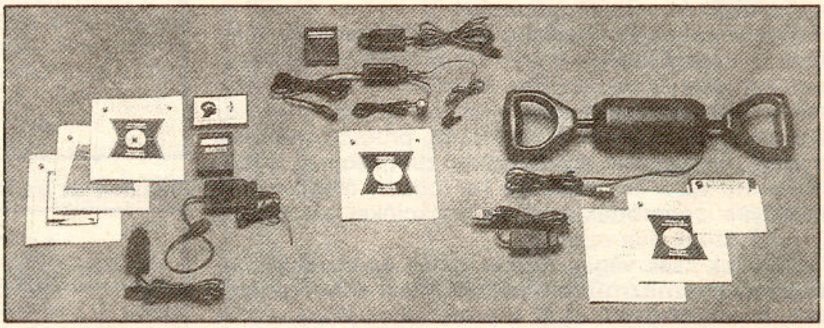
A kettes számú csomagban egy érzékelőt találunk, egy klipsz-szerű szerkezetet, amelyet a fülcimpára kell csippenteni! Ha a szívdobbanáskor a vérnyomás megemelkedik, ezt a szerkezet érzékeli, a bővítő portba helyezett modulon keresztül pedig jelet küld a számítógép felé.
Ha bekapcsoltuk a gépet, a modulban lévő „Ride for your life" program a résztvevő személytől néhány adatot kérdez – nem, életkor, testsúly, magasság stb. Ezek alapján fogja ugyanis az optimális terhelési adatokat kiszámítani.
Ezután máris kezdődhet a tréning. Lehet tornázni, szobabiciklizni, fekvőtámaszozni, vagy bármit csinálni, persze csak olyasmit, amihez elegendő a klipsz hosszabbító zsinórja. **A számítógép állandóan vigyázza és figyeli a terhelést, és az általa ideálisnak tartott mutatóktól való fölfelé vagy lefelé történő eltérésre fölhívja a figyelmet. Nem lehet tehát büntetlenül lazsálni!**
Hogy az edzés ne legyen unalmas, a programba itt is beépítettek egy játékot, amellyel az optimális terhelésnél lehet sikert elérni.

IZOMERŐSÍTŐ

A harmadik készlet két kézikönyvből, egy mágneslemezből – amely a szoftvert tartalmazza –, és egy, a joystick portra csatlakoztatható expanderből áll! Ez az utóbbi talán a legjobb ötlet!
Nyilván karizmainkat erősíthetjük vele. Erre szolgál a lemezen található „Harry Helio" című program is. A cél, hogy **a testi erő segítségével a helikoptert, a játékfiguránkat átvezessük az öszszes akadályon.** Minél kisebbre (azaz minél erősebben) nyomjuk expanderünket, annál magasabbra repül a helikopter. Már néhány perc múlva is lemérhetjük a tréninprogram izmainkra gyakorolt hatását – mondják a szerzők. Nyilván a nyelvünk kilóg az igyekezettől!

MENNYI?

Hogy ez-e a jövő útja, nem tudjuk. Az tény, hogy a Bodylog ezekkel a rendszerekkel teljesen új útra lépett, az ötlet eredeti és értelmes célt szolgál.
Talán az ár elrettent minket magyarokat, de a kinti piaci viszonyokhoz mérve azok inkább átlagosnak mondhatók. **A stresszprogram 199, a fitnesstréner 169, az izomerősítő 129 márkába kerül.** (Persze minden járulékos eszközt beleértve.) Az esetleges érdeklődők kedvéért a cég címeit is közöljük. Bodylog Inc., 34 Maple Avenue, Armonk, N. Y. 10504 USA Commodore Büromaschinen GmbH, Lyoner Str. 38, D-6000 Frankfurt am Main 71, c/o Bodylog Inc.





Programok GEOS-ban

II.

Cikksorozatunk eddigi részeiből már sok mindent sikerült megtudnia a figyelmes olvasónak a GEOS-ról. A GEOS programok készítéséről leírtak lehetőséget biztosítottak arra is, hogy betekintést nyerjünk a GEOS-ban történő programozás alapjaiba. Még most is az alapoknál tartunk, de elkészíthetjük első GEOS programjainkat.

Mindjárt az elején le kell szögezni a GEOS alatti programozás alapelvét. Ilyenkor tulajdonképpen assemblyben programozunk úgy, hogy közben a programozási munkánkat a GEOS rutinjainak felhasználásával végezzük. Így a rendelkezésre álló memóriát jól kihasználhatjuk, könnyen, gyorsan hozhatunk létre látványos, rövid és szép programokat.

A júniusi számban olvashattunk a GEOStory-ról. A leírtak is megerősítettek abban, hogy a GEOS voltaképpen egy igen jól, tervszerűen összeállított rutin-gyűjtemény. Másként fogalmazva: nem egy önálló, monolitikus program, hanem a strukturált programozás felé mutató, általánosan alkalmazható rutincsomag. Meghökkentő ez a kifejezés, de ha jobban megismerjük mennyire lebontották az egyes funkciókat, akkor nem csodálkozunk már rajta.

Gondoljuk csak végig: **külön rutin az ablakkészítéshez, a lemezműveletekhez, de a szövegkiíratás, a vonalhúzás, akár a pont kigyújtása rutint is elérhetjük.** Mindezek a programfejlesztés, tesztelés új útjait mutatják meg. Nem szükséges az egész programunkat egyben, egészében tesztelni, hanem kicsi, könnyen áttekinthető, egyszerű feladatokat elvégző rutinokat is tesztelhetünk. Ha ezek jók, akkor egymással való kapcsolatukat is kényelmesen ellenőrizhetjük. Nem kell megírni a billentyűzetlekérdező, kiíró, rajzoló stb. rutinokat, hanem paraméterezve használhatjuk ezeket a már kész modulokat. Az egyetlen szükséges dolog az, hogy tudjuk, milyen funkciót végeznek az egyes modulok és ehhez milyen paramétereket igényelnek. Most ezeket a rutinokat vesszük számba.

Először is a lemezen a futtatható file-t kell létrehozni. Ehhez több dolog szükséges. Igaz, a C= Újság 87/2-es számában már foglalkoztunk a lemezformátummal, de most újból visszatérünk rá, egy kicsit részletesebben. A GEOS alatt írt általános programot célszerű APPLICATION file-nak meghatározni. Így korlátlanul rendelkezhetünk a rutinokkal. Ha speciális (nyomtató vezérlő, bemenet vezérlő stb.) programot szeretnénk írni, ahhoz úgyis jobban bele kell mélyedni a GEOS-ba. Az APPLICATION file-típusnál a töltési, vég, ill. indítási (SYS) címet meg kell adni. A többi jellemző hiánya nem zavarja programunk futását.

ÚJABB CÍMEK

Először összefoglalva bemutatjuk a direktory felépítését a GEOS-ban:

Byte	Funkció
0	file típus
1,2	első adatblokk sáv és szektor címe
3-18	file név (SHIFT+SPACE feltöltéssel)
19-20	info blokk sáv és szektor címe
21	file felépítés
	0 - SEQential (CBM-DOS)
	1 - VLIR (GEOS-DOS)
22	GEOS file típus
	0 - nem GEOS file
	1 - BASIC
	2 - ASSEMBLER
	3 - adat file
	4 - rendszer file
	5 - segédprogramok
	6 - leírások
	7 - leírások adatai
	8 - FONT file
	9 - nyomtató-vezérlő
	10 - bemeti vezérlő, jelen esetben joystick
	11 - lemezmeghajtó vezérlő
23-27	a felírás dátuma, év/hó/nap/óra/perc formában
28-29	a blokkszám

Nézzük át az info blokk felépítését:

byte	funkció
0,1	a következő blokk címe (0,255-csak egy blokk van)
2,3	az ikon szélessége és hosszúsága (legtöbbször 24*21)
4	ikont definiáló byte-ok száma (63)
5-67	Ikon meghatározása, normál sprite formában.
68	Commodore filetipus 129-adat
	130-program
69	GEOS filetipus (ugyanaz mint a direktoryban)
70	filefelépítés 0-seqentiell
	1-VLIR
71,72	Program töltési cime
73,74	Program végcime (segédprogramoknál / 5 / van rá szükség)
75,76	inditási (SYS) cime a programnak
77-96	Program osztálya (CLASS)
97-116	Programozó neve
117-136	Adatfile-nál az előállító program neve (geoWrite, geoPaint)
137-159	Segédprogramok belső adatai
160-255	Infoszöveg (vége 0-val lezárva)

Ha valaki nem rendelkezik a geoMasterrel, akkor ezen táblázatok alapján előállíthatja a program megfelelő környezetét.

GEOS RUTINOK HASZNÁLATA

A GEOS-nak nagy a memóriaigénye a különféle rutinjai számára, azonban még így is igen nagy, 23 kilobytenyi terület áll a rendelkezésünkre ($0400-tól $5FFF-ig). A programozáshoz ésszerűbb és előnyösebb egy okos makroassemblert használni (például HYPRA-ASSEMBLER). Ekkor ugyanis strukturált, szép és könnyen áttekinthető programokat írhatunk. Sajnos az ilyen fordítók nálunk nem terjedtek el. A legtöbb C 64 tulajdonos egyszerű kétmenetes fordítóval dolgozik. Ezért a bárki számára elérhető **HELP+**-t választottam a forráslisták elkészítéséhez.

ELSŐ PÉLDAPROGRAM

Először a szövegkiíratással fogunk megismerkedni. Ennek a rutinnak a használatát az 1. példaprogramon keresztül fogom bemutatni. A feladat tetszés szerinti szöveg kiíratása a képernyőre úgy, hogy közben a GEOS ezirányú lehetőségeit minél jobban kihasználjuk. A szükséges rutinok paraméterei:

képernyőtörlés $CDA3	**nem kell paraméterezés**
szöveg kiíratása $C1AE	**jsr $C1AE**
	.word x koordináta 0–319
	.byte y koordináta 0–199
	.text a kiíratandó szöveg
	.byte 0 a szöveg vége
	0-val lezárva.

Ha végignézzük így a programot, máris láthatjuk, igen egyszerű. A különféle vezérlő és szövegpozicionáló kódokat az **1030–1180-as** sorokban definiáljuk. Az elnevezés egyértelmű, így csak a különlegességekre térek ki. A 20 érték (gotox) az x pozíciónak a következő szón (2 byte, alsó/felső byte formában) található értéket adja meg. Értéke: 0-tól 319-ig. (Ugyanis grafikus képernyőn dolgozunk, melynek a felbontása 320 × 200 pixel.) A 21 érték analóg módon a következő byte koordinátának tartalmát adja. (Értéke: 0-tól 199-ig.) A 22 érték (gotoxy) a kettő variációja, a vezérlőkódot követő szó tartalma x-et, az ezt követő byte tartalma y-t változtatja. A szövegkiíró rutint megfelelően paraméterezve a **1350-es** sorban hívjuk, majd körülbelül egyperces várakozás után a DESKTOP-ra térünk vissza. Visszatérés után észrevesszük, hogy a lemezen egy újabb file (SWAP FILE) jött létre. Piktogramja, adatai ugyanazok, mint az eredeti file-unknak. Ennek oka a betöltés módjában van. Az APPLIKATION file-ok betöltésénél egy segédfile – ez a SWAP FILE –, szükséges a rendszer számára, ennek törléséről nekünk kell gondoskodni. A megoldás az, hogy egyszerűen nem $C22C-vel térünk vissza a DESKTOP-ra, hanem a $918F rutinnal. Ez törli a már haszontalan SWAP FILE-t, és újraindítja a GEOS-t.

MÁSODIK ÉS HARMADIK PÉLDAPROGRAM

A visszatérésnek ezt a módját találjuk már a 2. példaprogramnál. Ezzel a rövid programmal végignézhetjük, milyen különböző hátterek választhatók az ablakoknál.

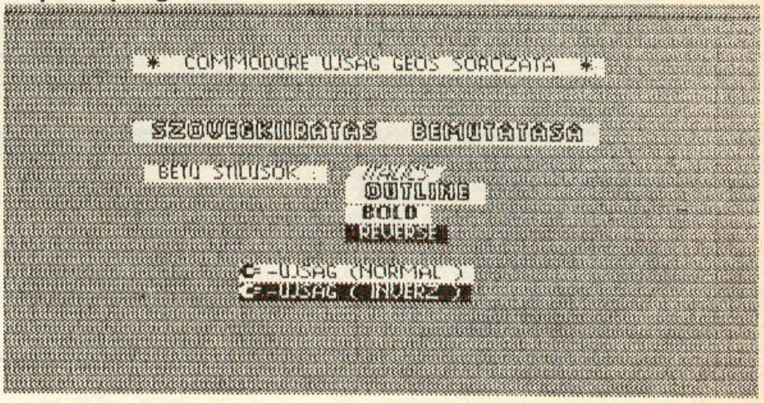
név	op kód	Paraméter	Funkció
OK	$01	.byte OK .byte x-pozició .byte y-pozició	Egy OK mezőt hoz létre amelyet "lenyomhatunk". Fontos hogy a beirt x-érék 8-al megszorozva addja a valóságos x-értékét az OK CANCEL stb mező bal felső sarkának.
CANCEL	$02	.byte CANCEL .byte x-pozició .byte y-pozició	CANCEL-mező
YES	$03	.byte YES .byte x-pozició .byte y-pozició	YES-mező
NO	$04	.byte NO	NO-mező
OPEN	$05	.byte OPEN .byte x-pozició .byte y-pozició	OPEN-mező
DISK	$06	.byte DISK .byte x-pozició .byte y-pozició	DISK-mező
DBTXTSTR	$0b	.byte DBTXTSTR .byte x-pozició .byte y-pozició .word mutató	Egy szöveg kiiratása a "mutatótól" az x/y pozicióra (x<255!) A szöveg végét 0-val zárjuk le!
DBVARSTR	$0C	.byte DBVARSTR .byte x-pozició .byte y-pozició .byte cim	Indirekt szövegkiirás Ugyanúgy működik mint a DBTXTSTR csak a szövegmutatót a nullás lapról a cim-ről olvassa be (cim, cim+1)
DBGETSTR	$0D	.byte DBGETSTR .byte x-pozició .byte y-pozició .by cim .by KAR.MAX	indirekt szövegbeirás a képernyő ablakszél x+xpozició/ ablakszél y+ypozició ról. ugyanúgy működik mint a DBVARSTR A cim-mel jelzett területet 0-val kell lezárni (addig kiirja az ottlevő szöveget) KAR.MAX a beolvasható karakterek számát határozza meg.
DBSYSOPV	$0e	nincs	egy köztes memóriahelyet fogla le, úgy ha a kijelölt mezőkon kivül nyomtak tűzgombot, akkor a $0e ill $851d-n levő cim ugrik.
DBUSROUT	$13	.WORD cim	A címtől kezdődő rutint végrehajtja az ablak kirajzolása után így van lehetőségünk pl: arra, hogy grafikát tehessünk az ablakba, még mielőtt a felhasználó bármit kiválasztott volna.

```
1000                 1          ;  A KIIRATAS ($C1AE) RUTIN BEMUTATASA           ;
1000                 2          ;
1000                 3          ;
0000                 4 NULLA   =0                        ;GEOS VEZERLOKOD TABLAZAT
0008                 5 BSPC    =8
0009                 6 FWSPC   =9
000A                 7 LFD     =10
000C                 8 UPL     =12
000D                 9 CR      =13
0012                10 RVSON   =18
0013                11 RVSOFF  =19
0014                12 GOTOX   =20
0015                13 GOTOY   =21
0016                14 GOTOXY  =22
001A                15 OUTLN   =26
0018                16 BOLD    =24
0019                17 ITALIC  =25
001B                18 PLAIN   =27
0080                19 COMM    =128
5000                20         *=$5000
5000 20 A3 CD       21         JSR $CDA3                 ; KEPERNYO TORLO RUTIN
5003 20 AE C1       22         JSR $C1AE                 ; KIIRATAS A MEGADOTT
5006 37             23         .BYTE 55                  ;X
5007 00             24         .BYTE 0
5008 23             25         .BYTE 35                  ;Y HELYRE
5009 20 2A 20       26         .TEXT " *  COMMODORE UJSAG GEOS SOROZATA  * "
502E 16             27         .BYTE GOTOXY
502F 37             28         .BYTE 55
5030 00             29         .BYTE 0
5031 46             30         .BYTE 70
5032 1A             31         .BYTE OUTLN
5033 20 53 5A       32         .TEXT " SZOVEGKIIRATAS  BEMUTATASA "
504F 0D             33         .BYTE CR
5050 0D             34         .BYTE CR
5051 1B             35         .BYTE PLAIN
5052 14             36         .BYTE GOTOX
5053 3C             37         .BYTE 60
5054 00             38         .BYTE 0
5055 20 42 45       39         .TEXT " BETU STILUSOK : "
5066 14             40         .BYTE GOTOX
5067 91             41         .BYTE 145
5068 00             42         .BYTE 0
5069 19             43         .BYTE ITALIC
506A 20 49 54       44         .TEXT " ITALICS "
5073 1B             45         .BYTE PLAIN
5074 1A             46         .BYTE OUTLN
5075 0D             47         .BYTE CR
5076 14             48         .BYTE GOTOX
5077 91             49         .BYTE 145
5078 00             50         .BYTE 0
5079 20 4F 55       51         .TEXT " OUTLINE "
5082 1B             52         .BYTE PLAIN
5083 18             53         .BYTE BOLD
5084 0D             54         .BYTE CR
5085 14             55         .BYTE GOTOX
5086 91             56         .BYTE 145
5087 00             57         .BYTE 0
5088 20 42 4F       58         .TEXT " BOLD "
508E 0D             59         .BYTE CR
508F 1B             60         .BYTE PLAIN
5090 12             61         .BYTE RVSON
5091 14             62         .BYTE GOTOX
5092 91             63         .BYTE 145
5093 00             64         .BYTE 0
5094 20 52 45       65         .TEXT " REVERSE "
509D 0D             66         .BYTE CR
509E 0D             67         .BYTE CR
509F 13             68         .BYTE RVSOFF
50A0 14             69         .BYTE GOTOX
50A1 64             70         .BYTE 100
50A2 00             71         .BYTE 0
50A3 80             72         .BYTE COMM
50A4 2D 55 4A       73         .TEXT "-UJSAG (NORMAL ) "
50B5 0D             74         .BYTE CR
50B6 12             75         .BYTE RVSON
50B7 14             76         .BYTE GOTOX
50B8 64             77         .BYTE 100
50B9 00             78         .BYTE 0
50BA 80             79         .BYTE COMM
50BB 2D 55 4A       80         .TEXT "-UJSAG ( INVERZ ) "
50CD 00             81         .BYTE 0
50CE A9 14          82         LDA #20                   ;IDOZITES ELOKESZITESE
50D0 A2 00          83         LDX #0
50D2 A0 00          84         LDY #0
50D4 88             85 IDOZIT  DEY                       ;VARAKOZAS
50D5 D0 FD          86         BNE IDOZIT
50D7 CA             87         DEX
50D8 D0 FA          88         BNE IDOZIT
50DA 38             89         SEC
50DB E9 01          90         SBC #01
50DD D0 F5          91         BNE IDOZIT
50DF 4C 2C C2       92         JMP $C22C                 ;VISSZA A DESKTOPRA
50E2                93         .END

ZEILEN:93   SYMBOLE:17   FEHLER:0

BOLD  =0018  BSPC  =0008  COMM  =0080  CR    =000D  FWSPC =0009  GOTOX =0014
GOTOXY=0016  GOTOY =0015  IDOZIT=50D4  ITALIC=0019  LFD   =000A  NULLA =0000
OUTLN =001A  PLAIN =001B  RVSOFF=0013  RVSON =0012  UPL   =000C
```

1. példaprogram






```
5000                         *=$5000
5000 A9 49                   LDA #$49        ; VISSZATERESI CIM
5002 8D 50 88                STA $8850       ; AMELYRE A SWAP FILE
5005 A9 CC                   LDA #$CC        ; VISSZAIRASA UTAN UGRANI KELL
5007 8D 51 88                STA $8851       ; ( $CC4A= RESET )
500A BA                      TSX             ; SP A VISSZATERESHEZ
500B 8E 52 88                STX $8852
500E A2 38           HUROK   LDX #$38        ;ABLAK DEFINICIO $5038-N KEZD
5010 A0 50                   LDY #$50
5012 86 02                   STX $02
5014 84 03                   STY $03
5016 20 56 C2                JSR $C256       ; ABLAK RUTIN
5019 EE 6B 50                INC DUMA2       ; BETU STILUS VALTOZTATAS
501C AD 6B 50                LDA DUMA2
501F C9 1C                   CMP #28         ; ERVENYES TIPUS PRINT
5021 D0 05                   BNE REKLAM
5023 A9 18                   LDA #24         ; KEZDI ELOLROL A TIPUSOKAT
5025 8D 6B 50                STA DUMA2
5028 A5 02           REKLAM  LDA $02         ; VALASZTOTT MEZO
502A C9 04                   CMP #$04        ; HA NO
502C F0 07                   BEQ KI          ; VEGE A PROGRAMNAK
502E EE 38 50                INC ABLAK       ; KOVETKEZO MINTA
5031 C9 20                   CMP #$20        ; ERVENYES MINTA $1F-NEL KISEBB
5033 D0 D9                   BNE HUROK
5035 4C 8F 91        KI      JMP $918F       ; SWAP FILE VISSZAIRASA+ RESET
5038 00              ABLAK   .BYTE 0         ; A MINTA
5039 32                      .BYTE 50        ; 50 FENTROL
503A 96                      .BYTE 150       ; 50 LENTROL
503B 32                      .BYTE 50        ; 50 JOBBROL
503C 00                      .BYTE 0
503D 13                      .BYTE 19        ; 75 BALLROL
503E 01                      .BYTE 1
503F 0B                      .BYTE $0B       ; SZOVEG KIRATAS
5040 25                      .BYTE $25       ; 37 BALLROL
5041 1A                      .BYTE $1A       ; ES 26 FENTROL
5042 50                      .BYTE $50       ; AZ 1.SZOVEG $5050-N KEZD
5043 50                      .BYTE $50
5044 01                      .BYTE 01        ; YES MEZO
5045 03                      .BYTE 03        ; BALLROL
5046 32                      .BYTE 50        ; FENNTROL
5047 04                      .BYTE 04        ; NO MEZO
5048 0C                      .BYTE 12        ; BALLROL
5049 32                      .BYTE 50        ; FENNTROL
504A 0B                      .BYTE $0B       ; SZOVEG KIIRATAS
504B 85                      .BYTE 133       ; 133 PIXEL BALLROL
504C 0A                      .BYTE 10        ; 10 PIXEL FENTROL
504D 6B                      .BYTE $6B       ; 2.SZOVEG $506B-N KEZD
504E 50                      .BYTE $50
504F 00                      .BYTE 0         ; ABLAK DEFINICIO VEGE
5050 20 CA 4F                .TEXT "  JOHET A KOVETKEZO MINTA ?"
506A 00                      .BYTE 0         ; SZOVEG VEGE
506B 18              DUMA2   .BYTE 24        ; BETU TIPUS
506C 80                      .BYTE 128       ; COMMODORE LOGO
506D 2D 55 4A                .TEXT "-UJSAG 1987 "
5079 00                      .BYTE 0         ; SZOVEG VEGE
507A                         .END

ZEILEN:56   SYMBOLE:5   FEHLER:0

ABLAK =5038  DUMA2 =506B  HUROK =500E   KI    =5035  REKLAM=5028
```

2. példaprogram

A C= újság 87/6-os számában már leírtuk a vezérlőkódok jelentését, így erre most nem térünk ki, csak a szükséges paraméterek táblázatát adjuk meg. Ezek alkalmazását láthatjuk a 3. példaprogramban.
Egyik ablakban bekérjük a nevet, a másikban pedig nyugtázhatjuk, hogy megfelelő-e a név. Érdemes továbbfejleszteni...

SEGÍTSÉG A BEVITELHEZ

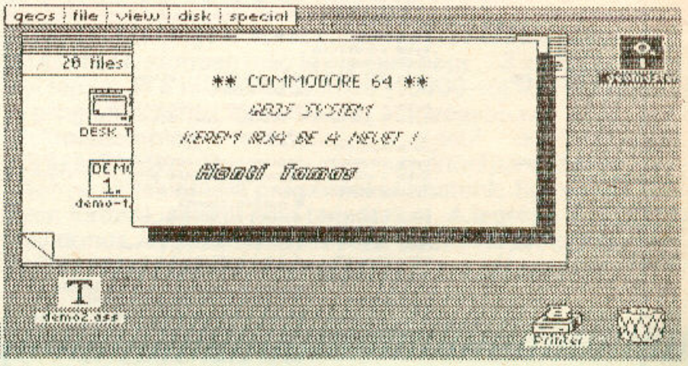
Számítva arra, hogy van, aki nem rendelkezik megfelelő HELP+ leírással, némi segítséget adunk az első lépések megtételéhez.

Miként állíthatjuk elő ezeket a GEOS alatt futó gépi kódú programokat?

Töltsük be a HELP+ programot, és **SYS 33000-rel** indítsuk el. Ezzel belépünk a HELP+ BASIC szerkesztőjébe, amely lehetőséget teremt arra, hogy a HELP+ assemblere részére **beírjuk a forráslistát.** Ha elkészültünk a begépeléssel, akkor **mentsük el** egy lemezfile-ba.
[*** lenyomásával** indítsuk el a fordítást. Az assembler három kérdésére az alábbiak szerint válaszoljunk:
1. adjuk meg a forrásprogram lemezfile nevét és RETURN
2. RETURN
3. 111 és RETURN

Ha nem jelzett hibát a fordító, akkor **mentsük ki** lemezre a gépikódú programunkat. Sajnos erre **nem nyújt segítséget a HELP+**, ugyanis a fordítás eredménye a tárba kerül, a forrásprogram által meghatározott helyre. A fordítási listából **meghatározható** a programunk első és utolsó utasításának címe. Ha a memória **$2B–$2C** címére beírjuk a program **első byte**-jának a címét, a **$2D–$2E** címére az **utolsó** byte-nál eggyel nagyobb címet (.end direktíva mellé írt cím a listában) akkor a BASIC programterület kezdetét és végét kijelző mutatókat állítottuk át. Ezután **SAVE utasítással** az éppen lefordított programunkat rögzíthetjük a lemezre. Ez már a tiszta gépi kódú rész, kis hibája, hogy a lemezen az első két byte-ban elhelyezi a töltési címet. Ez számunkra felesleges, hiszen nem normál Commodore DOS tölti, hanem a GEOS-DOS, az pedig az info blokkból veszi ki a töltési címet. Így alakul ki az a helyzet, hogy az így elkészített programnak **kettővel alacsonyabb címet kell adnunk,** hogy ne okozzon a memóriában eltolódást ez a két byte.
Következő lépésként a GEOMASTER segítségével **alakítsuk át** gépi kódú programunkat GEOS programmá, majd az info szektorban **adjuk meg** a kezdő, a vég és a start címet. Kezdőcím: $5000–2=20478, startcím: 20480, végcím: $5100=20736. Ha a végcím nem egyezik meg a valós végcímmel, az nem okoz problémát. Fontos viszont, hogy nagyobb vagy egyenlő legyen, mint ami a programnak kell.
Lemezmonitorral (pl.: EXDOCTOR, KWIK COPY stb.) **írjuk át** a direktoryban a file típusát 2-ről 5-re (assembly-ről application-ra). Ez a **22. byte**-on található. Azért van erre szükség, mert így segédprogramként a GEOS minden része benn marad a memóriában, és használhatjuk a rutinokat.
Ha kísérletezni szeretnénk, és nem akarjuk mindig ezt a meglehetősen hosszú eljárást végigcsinálni, elegendő egy jó applikációs file-t létrehozni. Ha az újabb programot szeretnénk kipróbálni, akkor elég az első adatblokk mutatóját (direktory file jellemzők 1., 2. byte-ja) a konvertált file-nál az újabb file mutatóira írni. Ha az indítási, illetve elhelyezési címek mások, csak akkor kell ezt az info szektorban megváltoztatni.

Honti Tamás



```
5000                      *=$5000       ; A PROGRAM AZ $5000 CIMEN KEZD
5000 A9 49                LDA #$49      ; VISSZATERESI CIM
5002 8D 50 88             STA $8850     ; AMELYRE A SWAP FILE
5005 A9 CC                LDA #$CC      ; VISSZAIRASA UTAN UGRANI KELL
5007 8D 51 88             STA $8851     ; ( $CC4A= RESET )
500A BA                   TSX           ; SP A VISSZATERESHEZ
500B 8E 52 88             STX $8852
500E A2 B2                LDX #$B2      ;BE CIME ($50B1)
5010 A0 50                LDY #$50
5012 86 0E                STX $0E
5014 84 0F                STY $0F
5016 A2 3D                LDX #$3D      ; 1.ABLAK DEFINICIO $503D-N KEZD
5018 A0 50                LDY #$50
501A 86 02                STX $02
501C 84 03                STY $03
501E 20 56 C2             JSR $C256     ; ABLAK RUTIN
5021 A9 B2                LDA #$B2
5023 85 0E                STA $0E
5025 A9 50                LDA #$50
5027 85 0F                STA $0F
5029 A2 59                LDX #$59      ; 2.ABLAK DEFINICIO $5059-N KEZD
502B A0 50                LDY #$50
502D 86 02                STX $02       ; ELTAROLASA
502F 84 03                STY $03
5031 20 56 C2             JSR $C256     ; ABLAK RUTIN HIVASA
5034 A5 02                LDA $02       ; VALASZTOTT MEZO
5036 C9 01                CMP #$01      ; HA OK
5038 F0 00                BEQ KI        ; VEGE A PROGRAMNAK
503A 4C 8F 91   KI        JMP $918F     ; SWAP FILE VISSZAIRASA+ RESET
503D 01         ABLAK1    .BYTE 1       ; A MINTA
503E 14                   .BYTE 20      ; 50 FENTROL
503F 7D                   .BYTE 125     ; 75 LENTROL
5040 3C                   .BYTE 60      ; 60 BALLROL
5041 00                   .BYTE 0
5042 FA                   .BYTE 250     ; 50 JOBBROL
5043 00                   .BYTE 0
5044 0B                   .BYTE $0B     ; SZOVEG KIRATAS
5045 25                   .BYTE $25     ; 37 BALLROL
5046 1A                   .BYTE $1A     ; ES 26 FENTROL
5047 6E                   .BYTE $6E     ; AZ 1.SZOVEG $506E-N KEZD
5048 50                   .BYTE $50
5049 0B                   .BYTE $0B     ; SZOVEG KIRATAS
504A 35                   .BYTE $35     ; 37 BALLROL
504B 2A                   .BYTE $2A     ; ES 26 FENTROL
504C 81                   .BYTE $81     ; AZ 2.SZOVEG $5081-N KEZD
504D 50                   .BYTE $50
504E 0B                   .BYTE $0B     ; SZOVEG KIRATAS
504F 15                   .BYTE $15     ; 37 BALLROL
5050 3A                   .BYTE $3A     ; ES 26 FENTROL
5051 8E                   .BYTE $8E     ; AZ 3.SZOVEG $508E-N KEZD
5052 50                   .BYTE $50
5053 0D                   .BYTE $0D     ; INDIREKT SZOVEGBEVITEL
5054 15                   .BYTE $15     ; BALLROL
5055 48                   .BYTE 72      ; FENNTROL
5056 0E                   .BYTE $0E     ; 0E-0F A CIM CIME
5057 15                   .BYTE 21      ; -MAX 21 KARAKTER
5058 00                   .BYTE 0       ; TABLA VEGE
5059                      ;
5059 0B                   .BYTE 11      ; 2.ABLAK DEFINICIO
505A 32                   .BYTE 50      ; 50 PIXEL FENTROL
505B 78                   .BYTE 120     ; 80 PIXEL LENTROL
505C 64                   .BYTE 100     ; 100 PIXEL BALROL
505D 00                   .BYTE 0
505E C8                   .BYTE 200     ; 100 PIXEL JOBBROL
505F 00                   .BYTE 0
5060 0B                   .BYTE $0B     ; SZOVEGKIIRATAS
5061 0A                   .BYTE 10
5062 14                   .BYTE 20
5063 A7                   .BYTE $A7     ;SZ2
5064 50                   .BYTE $50
5065 0C                   .BYTE $0C     ; INDIREKT SZOVEG KI
5066 0A                   .BYTE 10
5067 23                   .BYTE 35
5068 0E                   .BYTE $0E
5069 01                   .BYTE 01      ; OK MEZO
506A 03                   .BYTE 03
506B 32                   .BYTE 50
506C 00                   .BYTE 0       ; TABLAZAT VEGE
506D 18         SZ1       .BYTE 24
506E 2A 2A 20             .TEXT "** ¬OMMODORE 64 **"
5080 00                   .BYTE 0
5081 19                   .BYTE 25
5082 C7 C5 CF             .TEXT "I T♥ SYSTEM"
508D 00                   .BYTE 0
508E CB 45 52             .TEXT "╯EREM IRJA BE A NEVET !"
50A5 1A                   .BYTE 26      ; OUTLINE
50A6 00                   .BYTE 0       ; SZOVEG VEGE
50A7 D5 44 56   SZ2       .TEXT " ╭DVOZLOM :"
50B1 00                   .BYTE 0       ; SZOVEG VEGE
50B2 20         BE        .TEXT " "
50B3 00                   .BYTE 0
50B4                      .END

ZEILEN:92    SYMBOLE:5    FEHLER:0

ABLAK1=503D  BE    =50B2  KI    =503A  SZ1   =506D  SZ2   =50A7
```

3. példaprogram

RAJZOLÓPROGRAMOK



MORVAI LÁSZLÓ
25 éves, biológus mérnök, a BME Mezőgazdasági Kémiai Tanszékének tudományos segédmunkatársa



TÓTH ZOLTÁN
26 éves, grafikus, szellemi szabadfoglalkozású



NYISZTOR ANDOR
14 éves, a Thököly Gimnázium tanulója



TALLÉR JÓZSEF
26 éves, a Commodore újság szerkesztője

Ismét csak rajzolóprogramok futtatására adtuk a fejünket, csakhogy ezúttal – az elmúlt havi vizsgálódásainkkal ellentétben – nem a C–64-re, hanem a Plus/4-re készített grafikai programokkal foglalkoztunk. Amíg a C–64-es rajzolóprogramokat kis túlzással elavultnak nyilvánítottuk – mondván, hogy a vérbeli programozók ezek helyett már Atari-szoftvereken dolgoznak –, addig a Plus/4-es rajzolóknak mind a mai napig hatalmas szerepe van azok szoftver-tárában is, akik hivatásszerűen készítenek programokat erre a gépre.

Nagyon nehéz volt a futtatható programok összegyűjtése. A C–64-nél a bőség zavarával küszködtünk, most viszont majd hogy nem arra fanyalodtunk, hogy egyetlen programot állítsunk a starthoz, és azt önmagával versenyeztessük – amit nyilván mindenki nevetségesnek ítélne. Végül is azért sikerült összeszednünk három használható szoftvert – de ismét csak kérjük olvasóink segítségét: aki tud még Plus/4-re készült színvonalas rajzolóprogramról, az értesítse szerkesztőségünket!

Még egy komoly eltérés az előző programfutamtól: a most futtatott programok többsége – könnyen kitalálható: a három közül kettő – magyar gyártmány, így a kereskedelemben is beszerezhető.

Nevezzük végre meg a rajtoló programokat! A **BOTTICELLI** a **Novotrade** terméke, ez a 2C üzletekben kapható. A **TED PAINT**-et az AIRIA Software Group készítette (félreértés ne essék: ez is magyar szoftver), – még sehol nem árulják, de legújabb információink szerint a **Fotoelektronik – Novotrade GT**-nél már megrendelhető. Harmadik, futtatott programunk, a **PAINT BOX** viszont angol gyártmány.

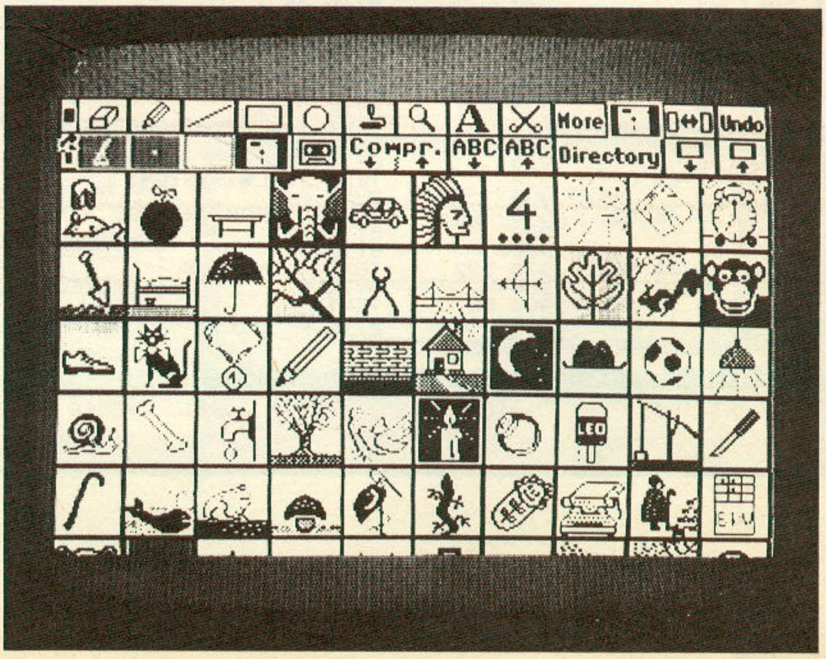
Azt is meg kell említenünk, hogy a **Paint Box** C–16-os gépen is fut, azaz memóriakihasználása gazdaságos, viszont kevesebb funkció végrehajtására képes, mint társai – de erre még visszatérünk a szubjektív vélemények kapcsán.

A 64-es programok esetében kitértünk arra, hogy azok általában kalózmásolás útján terjednek. Így most is meg kell említenünk, hogy a **Paint Box**-nak is ez a sorsa. A **Ted Paint** szintén másolható – bár sokkal keservesebben. A **Botticelli** védelmét viszont nagyon rafináltan oldotta meg a Novotrade: a program csak akkor futtatható, ha a számítógép lemezmeghajtó dugaszába egy külön csatlakozóelemet illesztünk. Ha a programot lemezről töltjük be, akkor ezt a csatlakozót a számítógép és a lemezmeghajtó közé kell tennünk – ha pedig kazettáról, akkor ez a kis hardver-dugó csak „fityeg" önmagában. Enélkül azonban a kép elkásásodik, vagy összevissza váltakoznak rajta a színek. (A kézikönyv a csatlakozót mint „hengerded tárgy"-at említi, a Novotrade-nél viszont csak mint **műfa...-t** emlegetik. Mit mondjunk, ez utóbbi név jobban illik rá.) Ez a csatlakozó csak a **Botticellivel** együtt kapható – így aztán a programfeltörésnek nincs sok értelme.

Amit még fontos megemlítenünk: a **TED Paint** program sok tekintetben a C–64-re készült **Doodle** átiratának tekinthető. Ezt nem elmarasztalásként közöljük, sőt! – hiszen a Doodle egy igen jól használható rajzolóprogram, és csak örülhetünk, hogy van Plus/4-es változata is. Csupán azért tartottuk szükségesnek megjegyezni, hogy a továbbiakban senki ne lepődjön meg, ha a **TED Paint** és a múltkoriban bemutatott **Doodle** ismertetésében sok azonos vonást talál.

Mivel szerény kis „stábunk" nem bővült – sőt létszáma csökkent – az előző programfutam óta, így vitánk még kevesebb volt az akadályokról, mint akkor. Vagyis három rövid mondatban megegyeztünk, hogy az akadályok maradjanak ugyanazok, mint legutóbb. Azért persze ezeket minden futamnál röviden újra ismertetjük, hiszen elképzelhető, hogy a Plus/4 tulajdonosok közül sokan nem olvasták el előző futtatásunk eredményét.

Újra csak a már bevált „bővített táblázatok módszerét" alkalmaztuk az értékelés során: azaz egy-egy részakadálynál nem csak plusz, vagy mínusz jellel jelöltük, hogy a program képes-e annak teljesítésére, hanem rövid megjegyzésben azt is megadtuk, hogy milyen szinten tudja ellátni a szolgáltatást.



1. AKADÁLY: ALAPFUNKCIÓK

Hogyan tudja kirajzolni a program a legelemibb geometriai alakzatokat?

RAJZOLÓPROGRAMOK

	Botticelli	TED Paint	Paint Box
Szabadkézi rajz	jó	jó	jó
Egyenes szakasz	jó	közepes	jó
Törött vonal	jó	jó	jó
Kör	jó	elfogadható	jó
Téglalap	jó	közepes	jó
Színállítás	közepes	jó	gyenge
Nagyítás (javítás)	gyenge	közepes	nincs
Feltöltés színnel	jó	jó	jó

● Szabadkézi rajz

Ez a legelemibb szolgáltatás, erre természetesen mindegyik program képes. Hogy azután ez mennyire könnyen használható, arra majd a „Kezelhetőség" akadálynál térünk vissza.

● Vonal, töröttvonal

Ismét csak azt kell mondanunk, hogy egyenes vonalat valamennyi program tud rajzolni. A „hogyan"-ban azonban már jelentős eltérések mutatkoznak. A **TED Paint**-nél beállíthatjuk a vonalszakasz két végpontját, de a program nem mutatja meg, hogy a kész vonal hogyan is fog kinézni. Így azután lehetetlen például olyan vonalat húzni, amely érint egy, már megrajzolt kört, vagy átmegy egy, már meglévő ponton.

A **Botticelli** és a **Paint Box** viszont egy – általunk „gumizásnak" nevezett – eljárás révén képes arra, hogy villogva mindig megmutassa a két kijelölt végpont közötti vonaldarabot. Sőt, a két programnál lehetőségünk nyílik arra is, mint a C–64-en megismert Koala Painternél: azaz, ha megrajzolunk egy vonalat – vagy bármilyen más alakzatot –, azt még nem kell feltétlenül a rajzolás helyén hagynunk, hanem tetszőlegesen áthelyezhetjük, és ott rögzíthetjük a képernyőn, ahol akarjuk.

Nézzük most a töröttvonalat: azaz tud-e a futtatott szoftver a végpontjaiknál egymáshoz illeszkedő egyenes szakaszokból álló, folyamatos vonalat húzni. A TED Paint – a Doodle mintájára – képes rá, a **Botticelli** szintén, és a **Paint Box**-nak is külön menüpontja van erre.

● Kör, ellipszis

Ellipszist csak a **TED Paint** tud rajzolni – az eljárás során először a középpontot kell megadnunk, majd vízszintes és függőleges irányban tetszőlegesen tágítható az alakzat. Egy megjegyzést azonban feltétlenül kell tennünk ezzel kapcsolatban: szép-szép, hogy egy programmal lehet ellipszist rajzolni –, viszont pontos kört így csak nehezen hozunk létre, igen jó szemmérték kell hozzá. A másik két program csak kör rajzolására képes: először egy kerületi pontot, majd pedig a középpontot kell megadnunk. A Botticelli nagy hátránya, hogy a választott kerületi pont csak a körvonal legalsó, vagy legfelső pontja lehet.

Ennek ellenére a kirajzolt görbék mindhárom program esetében szépek, tiszták, a **Botticelli** és a **Paint Box** körei nem torzak.

● Téglalap

Ezt a funkciót is ismeri mindhárom program. A Botticelli és a Paint Box esetében a két átellenes sarkot kell megadnunk – persze, e két programnál a kész téglalap „továbbcipelhető". A **TED Paint**-nél – és itt ismét csak a Doodle-ra kell hivatkoznunk – a körhöz hasonlóan a téglalap középpontját adjuk meg, majd innen kiindulva tágítható és szűkíthető a szélesség, illetve a magasság.

● Színállítás, színváltás

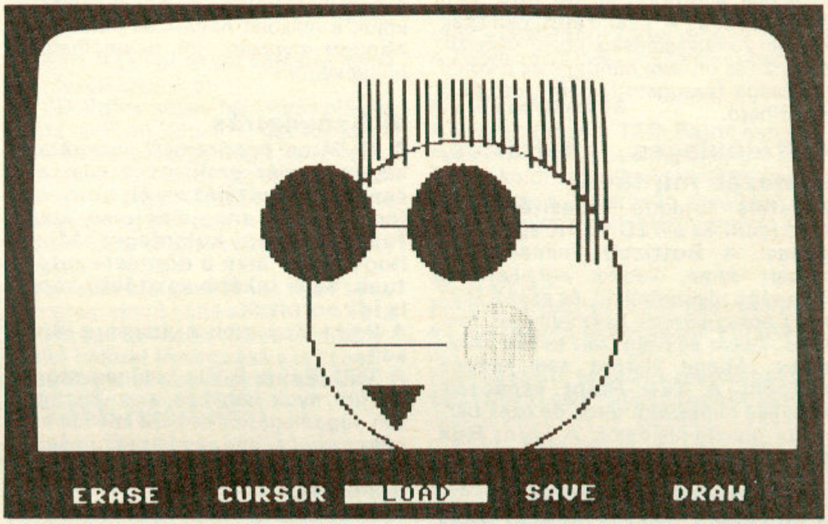
Az alap és a háttér színét minden programnál még a kezdet kezdetén beállíthatjuk. A **Paint Box**-nál létezik ugyan színváltás rajzolás közben is – de ez igen nehezen érhető el, érteni kell hozzá. A **Botticelli**-é szintén keserves, de a **TED Paint**-nél nagyon jól működik. Itt bármilyen papír- és tintaszín kombinációt könnyen beállíthatunk rajzolás közben is – csakhogy ilyenkor a már meglévő rajzunkat karakterhelyenként színezhetjük csak át. Efféle lehetőséggel – sokkal nehezebben elérhető módon – rendelkezik a **Botticelli** is. Ugyanis először törlés (radír) üzemmódba kell lépnünk, majd itt beállítani a tinta- és papírszínt, és ezután végigsatírozni a megfelelő képernyőrészt.

● Nagyítás, javítás, finom rajz (Zoom)

A **Botticelli** képes arra, hogy nyolcszor nyolcas nagyításban jelenítse meg a képernyő kiválasztott részletét, és itt javíthatunk is. Ezen belül van törlő és invertáló üzemmódja is. A **TED Paint** szintén nyolcszorosára képes nagyítani a rajz részleteit, és itt szinte bármit megtehetünk, mint amit a normál képernyőn – azaz bármilyen rajz-üzemmódot használhatunk, színt állíthatunk, törölhetünk, másolhatunk. A **Paint Box** viszont egyáltalán nem tud nagyítani – így itt erről a funkcióról nem is beszélhetünk.

● Feltöltés színnel

Erről nincs sok mindanivalónk: a funkció teljesítésére mindhárom program képes, méghozzá igencsak tisztességes sebességgel.



	Morvai	Tóth	Nyisztor	Tallér	Átlag
Botticelli	4,5	4,8	4,5	4,5	4,58
TED Paint	5	4,6	5	4,5	4,78
Paint Box	4	3,6	4	4	3,90

Az alapfunkciók értékelése után most is szoros maradt a mezőny – akár a C–64-es programok értékelésekor –, bár most kissé jobban lemaradt „szegény jó" Paint Box. Hogy mi indokolja ezt az idézőjelbe tett kettős jelzőt?

Szegény, mert sok programfunkciót csak körülményesen, és nem is igazából képes végrehajtani. Jó viszont, mert tudomásunk szerint ez az egyetlen C–16-os rajzoló, ami kísérletet tesz egy elfogadható minőségű rajz előállítására.

2. AKADÁLY: KÜLÖNLEGES FUNKCIÓK

Nézzük meg most, hogy mi olyan szolgáltatásai lehetnek még egy rajzolóprogramnak, amelyek ugyan nem alapfunkciók – de mégis sokat segíthetnek egy jó rajz elkészítésében!

	Botticelli	TED Paint	Paint Box
Sugár	jó	gyenge	jó
Teli kör, téglalap	közepes	jó	jó
Toll és ecsetforma változtatása	közepes	gyenge	gyenge
Színezés mintával	jó	nagyon jó	nincs
Színcsere, invertálás	nagyon jó	gyenge	nincs
Ablakműveletek	jó	közepes	gyenge
Tükrözés, forgatás	nagyon jó	gyenge	nincs
Szövegírás	nagyon jó	jó	nincs
Előzetes megjelenítés	nagyon jó	jó	nincs
Egyéb	memória	hibavédelem	nincs

● Sugárrajzolás
Lehet-e egy előre megadott pontból tetszőleges számú egyenes szakaszt húzni?
A **Botticelli** és a **Paint Box** tudja ezt, a **TED Paint**-nál is megoldható, nehézkesen.

● Színnel feltöltött kör, téglalap
Ezt az igényünket talán sokan túlzásnak tarthatják. Sok esetben kényelmes azonban, ha egy geometriai alakzatot a gép igényeink szerint rögtön színnel feltöltve rajzol ki.
A **Botticelli**-nél nincs erre külön funkció – bár kis ügyeskedéssel elérhető. A **TED Paint** és a **Paint Box** ezt közvetlenül meg tudja tenni.

● Toll vagy ecset formájának, színének változtatása
Ezalatt azt értjük, hogy vajon megtehető-e, hogy egy-egy rajzrészletet más vonalvastagsággal – esetleg másféle ecsetnyommal – készítsünk el, mint a többit.
A **Botticelli** tizenhat ecsettel rendelkezik, ezek jól használhatók, – de nem tervezhetünk „saját használatú" ecsetet. A **Paint Box**-nak szintén van nyolcféle ecsetformája, de ezek sem tervezhetők át. A **TED Paint**-ben csak kétféle vonalvastagság közül választhatunk (és ott sem mindig), de a vonal minősége (szaggatott, pontozott stb.) kijelölhető.

● Különleges feltöltés, színezés mintával
Ilyesféle funkció teljesítésére a Botticelli és a TED Paint egyaránt képes. A **Botticelli** mintakészlete hatvan darab, kétszer két karakter nagyságú alakzatból áll, és ezeket bármely ecsetmérettel használhatjuk festésre, illetve egy-egy zárt terület kitöltésére. Mintát viszont nem tudunk megadni. A **TED Paint** mindössze négyféle mintázatot tárol, de ezek bármikor újradefiniálhatók. A **Paint Box** sajnos ismét csak csődöt mondott.

● Színcsere, invertálás
A **Botticelli** mindkét funkció gyors végrehajtására képes. A **TED Paint** képes az invertálásra, de nehézkesen. A **Paint Box** egyiket sem tudja.

● Ablak-műveletek
Azt néztük meg, hogy a programokban kezelhető-e önállóan a képernyő egy-egy részterülete.
Hát, ez bizony nem erőssége egyik vizsgált programunknak sem. Egy kijelölt tartomány másolására képes mindhárom program – a **Paint Box** ennél többet nem is tud. A **TED Paint**-nél megválasztjuk: a terület megmaradjon-e az eredeti helyen, vagy törlődjön, illetve, a másolat megjeleníthető inverzben is.

A **Botticelli** mindezen túl képes az adott képernyőtartomány nagyítására, kicsinyítésére, sőt, egyéb extra szolgáltatásokra is, amelyeket a következő funkciónál részletezünk.

● Tükrözés, forgatás
A **Paint Box** nem tud semmi ilyesmit.
A **TED Paint** is meglehetősen gyenge e téren: csak a teljes képernyő vízszintes és függőleges középvonalára, valamint középpontjára képes tükrözni az egész rajzot. A **Botticelli** a képernyő bármely kijelölt részterületét tudja akár vízszintesen, akár függőlegesen, akár középpontosan tükrözni, sőt 90 fokonként forgatni is. Ezen túl képes a képernyőterületet úgy is másolni, hogy megadjuk a másolat háttér- és tintaszínét. Mindezt gyorsan, jól használhatóan hajtja végre.

● Szövegírás
A C–64-es programok vizsgálata kapcsán már említettük tesztelő csoportunk vitáját arról, hogy vajon a betűk megjelenítése alapfunkció-e, vagy különleges? Minthogy akkor arra a döntésre jutottunk, hogy inkább az utóbbi, most is ide soroltuk.
A **Paint Box** ezen a vizsgán is elbukott.
A **TED Paint** 8-féle betűnagysággal tud írni, nyolc irányban, azaz vízszintesen, függőlegesen, és a két átló mentén oda-vissza. A programírásnál megszokott módon cserélhető a kétféle karakterkészlet is.
A **Botticelli** szintén képes normál és mindkét irányban egytől négyig nagyított méretű betűk megjelenítésére, és a szöveg négy irányban írható. Szövegírás közben is bekapcsolható a tükröző funkció – azaz a kiírt karakterek tükörképe jelenik meg. A program rendelkezik saját karaktergenerátorral is, ami könnyen elérhető és jól használható, így akár a teljes betűkészlet átírható. A már leírt szöveg javítása viszont rendkívül nehézkes.

● Előzetes megjelenítés
Korábban már említettük, hogy „gumizó" funkciója csak a **Botticelli**nek van. Egy alakzatot nem kell rögtön a végleges helyén megrajzolnunk, a program lehetőséget nyújt az „átszállításra". Ezen túl hatalmas előny még a két rajzlap: az egyik lapot használhatjuk munkaterületnek, és a végleges részleteket átvihetjük a másik lap megfelelő helyére. A **Botticelli** keretmenüt használ, azaz a kép felső részének egy csíkjában olvashatók a választható funkciók. Így ez a sáv eltakarja a rajz egy részét – de van egy olyan utasítás, amelynek segítségével a menüsáv vastagságának megfelelően lejjebb vagy feljebb tolhatjuk a rajzot.
A **TED Paint** előzetes rajzmegjelenítési funkciói inkább a kezelhetőség körébe tartoznak, így annak vizsgálatakor vesszük sorra őket – de túl sok jót senki ne várjon. A **Paint Box** pedig ismét csak nem tud ilyen elvárásoknak eleget tenni.

● Egyéb
Nos, milyen „extrái" vannak még programjainknak az eddigieken kívül?
A **Paint Box**-nak, sajnos, semmilyen.
A **Botticelli** tud még egy-két különleges dolgot – de azokról inkább a tárolás kapcsán szólunk.
A **TED Paint**-nek viszont nagy érdeme, hogy védi a felhasználót az akaratlan rajzmódosításoktól. Szinte minden utasítás végrehajtása előtt megjelenik a „Biztos benne?" kérdés, és a parancs csak igenlő válasz esetén hajtódik végre.

	Morvai	Tóth	Nyisztor	Tallér	Átlag
Botticelli	5	4,8	4,5	4,5	4,70
TED Paint	4,5	4,4	4,5	4,48	4,48
Paint Box	2,5	3	2,5	3	2,75

3. AKADÁLY: KEZELHETŐSÉG

Itt a menük érthetőségét, egyértelműségét, a rajzkezelés könnyedségét vizsgáltuk, és azt, hogy az egyes menüpontok milyen könnyen érhetők el.

	Botticelli	TED Paint	Paint Box
Menük egyértelműsége	világos	jó	közepes
Menük hozzáférhetősége	közepes	jó	közepes
Help	nincs	nincs	nincs
Undo	jó	gyenge	nincs
Sebesség	gyenge	állítható, jó	nagyon jó
Háló	jó	gyenge	nincs
Több munkalap	közepes	nagyon jó	nincs
Koordináta kiírás	gyenge	nincs	nincs
Nyomtatás	jó	nagyon jó	nincs
Egyéb	nagyon jó vezérlés	zavaros irányítás	elfogadható kezelés

● Menük egyértelműsége:

A **Botticelli** ikon-menüvel dolgozik – kis ábrácskák jelzik, hogy melyik szolgáltatás mire jó. Az almenükben is mindig rajzok segítenek az eligazodásban. A **TED Paint** sajnos sokat átvett a Doodle menükezelésének stílusából – így ez is gyakran áttekinthetetlen. A legérdekesebb a **Paint Box** menükezelése. A képernyő alján egy keskeny sávban jelennek meg a menüpontok, és a sávot, ide-oda húzogathatjuk – sőt, akár körbe is. Ha kiválasztjuk a szükséges funkciót, akkor annak neve inverzbe vált. Primitívnek tűnik a módszer, de rendkívül jól használható ilyen kevés menüpontot használó programnál.

● Menük hozzáférhetősége

Mindhárom program menüje és almenüjei is viszonylag könnyen elérhetőek. „Viszonylag", hiszen a **Paint Box** csak egyetlen alapmenüvel rendelkezik, aminek aktuális része mindig olvasható a képernyő alján, – az viszont nehézkessé teszi használatát, hogy új funkció választásakor a menüsort mindig körbe kell görgetnünk.

A **Botticelli** keretmenüje, és valamennyi almenüje egyértelmű, de egy almenüből nehéz kikeveredni.

A **TED Paint**-ben külön billentyűkombinációja van mindegyiknek, sőt, a kilépésnek is.

● HELP és UNDO

Olyan utasítás, amelyet egyértelműen HELP-nek nevezhetnénk, egyik programban sincs. Mit tehetünk hát, ha már annyira belezavarodtunk valamelyik programba, hogy nem tudjuk a továbblépést?

A **Paint Box** esetében a válasz: semmit. Vigasztaló viszont, hogy a menü mindig olvasható a képernyő alján, így különösebben nem tévedhetünk el az utasítások között.

A **TED Paint** főmenüje elérhető egy billentyűkombináció lenyomásával, innen az almenük a már ismert módon hívhatóak. A legrosszabb a helyzet a Botticellinél: a menürendszer nem ad lehetőséget arra, hogy a főmenüt bármikor elérhessük.

Térjünk rá az UNDO funkcióra!

Ez az utasítás azt jelenti, hogy a program a legutóbbi parancsot semmisnek veszi, annak hatását törli.

A **Paint Box** szurkolóinak újabb rossz hír: a program ezt a funkciót sem tudja. A **Botticelli**nél nagyon jól működik, a **TED Paint**-en viszont meglehetősen furcsán. Itt ugyanis egy rajzolási művelet előtt a meglévő ábrát el lehet helyezni a gép memóriájába. Ha az új parancs elrontja a képet, akkor visszahívhatjuk az előzőt. Igen ám, de ki tudja előre, hogy melyik utasítás lesz kockázatos? A funkciónak emiatt nincs túl sok értelme – csak akkor, ha minden új parancs előtt eltároljuk a képet; ez viszont rendkívül időigényes.

● Sebesség:

A **Botticelli**nél nem tudjuk állítani a rajzoló kurzor sebességét – de a programnak megvan az a szolgáltatása, hogy vonalhúzáskor a szálkereszt fokozatosan gyorsul fel.

A **TED Paint** többre képes: a sebesség megválasztható 1 és 9 között. Ráadásul egyetlen billentyű lenyomásával a rajzoló kurzor a képernyő bármelyik sarkába, illetve középpontjába helyezhető.

A **Paint Box**-nál szintén állítható a gyorsaság, nagyságát egy fehér sáv jelöli, amit rövidíthetünk vagy hosszabbíthatunk.

● Háló, koordinátakijelzés:

A **Paint Box**-ról ennek kapcsán ismét csak felejtkezzünk el.

A **TED Paint** képes háló kirajzolására – ami sokban segíti a tájékozódást. Csak az a probléma, hogy ha felrajzolunk egy hálót, és így kezdünk dolgozni, majd letöröljük a rácsozatot, akkor a háló önmagával együtt törli azokat a rajzpontokat, amelyek a rácsot alkotó vonalakon voltak.

A Botticelli esetében nincs ilyen hiba, a program eltérő háttérszínekkel vetít rá a rajzra egy sakktábla-hálót. Sőt, kényszerítő hálózat létrehozására is képes – de erről is a tárolás kapcsán szólunk.

● Több munkalap

A **Paint Box**-ról e funkció kapcsán most sem tudunk mit mondani.

Mint már említettük, a **Botticelli** rendelkezik két képernyővel.

A **TED Paint**-nek viszont még több jó szolgáltatása van. Lehetőséget biztosít arra, hogy a képernyő tetszőleges területét elhelyezzük egy köztes tárolóba, majd pedig szükség szerint visszahívjuk. Megválaszthatjuk, hogy a színeket is tároljuk-e, vagy csak a grafikát; hogy a memóriában tárolt képet milyen VAGY-kapcsolattal tegyük az új helyre; stb. Ha egy képernyőrészt így el tudunk tárolni, akkor az úgy tekinthető, mint különálló munkaterület – azaz munkalap. Ehhez persze hozzá kell fűznünk azt, hogy a **TED Paint** nehézkesen kezeli ezeket a külön tárolt területeket. Ügyeskednünk kell ahhoz, hogy a program azt hajtsa végre, amit mi akarunk – de megéri.

● Koordináta-kiírás

Erről nincs sok mondanivalónk: csak a **Botticelli** képes erre, az is csak a Zoom (nagyítás) – üzemmódban.

● Nyomtatás

A **Botticelli** és a **TED Paint** remekül működik az MPS nyomtatókkal. A **TED Paint** különleges szolgáltatása: képes arra, hogy a képernyőn megjelenített rajzot a nyomtatón négyszeresére nagyítsa.

● Egyéb

Amiről érdemes még beszélnünk, az a kezelés eszközeire vonatkozik.

A **Botticelli** egyaránt megbízhatóan vezérelhető csak billentyűzetről, illetve csak joystick-ról is. Hasonló a helyzet a **Paint Box**-nál is. A **TED Paint** a kétféle vezérlés keverékét igényli, ami igencsak ellenszenves. A menü és az almenü pontjait a billentyűzetről kell kiválasztani – bár időnként itt is segít a botkormány –, utána pedig, rajzolás közben csak a joysticket lökdöshetjük.

	Morvai	Tóth	Nyisztor	Tallér	Átlag
Botticelli	5	4,6	4,5	5	4,78
TED Paint	3	3,3	3,5	4	3,46
Paint Box	4,5	3,8	4	4	4,08

RAJZOLÓPROGRAMOK

4. AKADÁLY: TÁROLÁS

Már korábban is megállapítottuk, hogy minden rajzolóprogramnál alapvető követelmény a kész képek tárolása. A kérdés csak az, hogy ezt tudják-e kazettán is, vagy csak lemezen, valamint, hogy mit tudnak még ezen kívül?

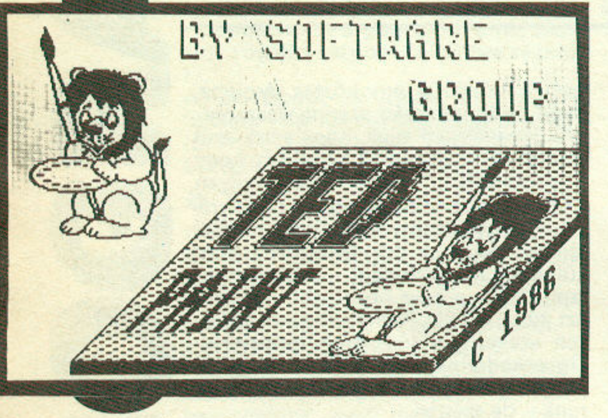
	Botticelli	TED Paint	Paint Box
Tárolás lemezen	jó	jó	jó
Katalógus	nagyon jó	jó	nincs
Verify	nincs	nincs	nincs
Törlés	nincs	gyenge	nincs
Törlés védelem	nincs	nincs	nincs
Kazetta	nagyon jó	jó	jó
Egyéb	hálózat, tömörítés	nincs	nincs

● Tárolás lemezen

Ez elengedhetetlen funkció, így azután ezt teljesíteni tudja mindhárom program.

● Katalógus

A **Botticelli** a directory kérése után kilistázza az összes, lemezen lévő file-t, az általa készített rajzok nevét eltérő színnel emeli ki. A **TED Paint** csak azokat az állományokat (rajzok nevét) listázza ilyenkor, amelyeket ezzel a rajzolóprogrammal készítettünk. A **Paint Box** itt ismét csak csődöt mond.



● Verify

A frissen felvett adatállomány azonnali ellenőrzésére sajnos – a C–64-es szoftverekhez hasonlóan – egyik mostani program sem alkalmas. Pedig nem lenne haszontalan szolgáltatás, sőt.

● Törlés

A **Paint Box**-nak ismét csak nincs erre külön utasítása – de a **Botticelli**nek sincs. A **TED Paint** esetében pedig a Command funkció választása után a lemezegységnek küldhetünk parancsokat, annak saját szabályai szerint – azaz törölhetünk, átnevezhetünk, formázhatunk – feltéve, hogy ismerjük ezeket a parancsokat. Vagyis: e funkció teljesítésében sem jeleskedtek a vizsgált programok. Elvégre egy rajzszoftver használójától nem kellene elvárni, hogy ismerje a lemezegység utasításait, mint ahogy az a **TED Paint** teszi; a másik két programnak pedig eleve nincs olyan parancsa, ami ezt a funkciót teljesítené.

● Törlésvédelem

Ez aztán végképp hiányzik a vizsgált programokból.

● Kazetta A

Botticelli nagyon barátságosan kezeli a kazettát: eleve abból indul ki, hogy a program használója kazettás egységgel dolgozik, így a szoftver az ő szempontjaihoz igazodik. Emellett, persze, képes lemezegység kezelésére is (sőt, a korábban leírt katalóguskészítést csak itt tudja elvégezni). Az mindenesetre figyelemre méltó, hogy végre találtunk egy olyan programot, amely a kazettahasználókat részesíti előnyben. A **TED Paint** és a **Paint Box** is egyaránt képes kazettán és lemezen tárolni az elkészült rajzokat. Ez a **Paint Box** esetében figyelemre méltó, mert csak dicsérni lehet, hogy a szerzőknek ilyen kis memóriaterjedelem mellett effélére is kiterjedt a figyelmük.

● Egyéb

Nos, itt kell rátérnünk arra, hogy tárolás terén a **Botticelli** különleges, máshol nem tapasztalt szolgáltatásokkal rendelkezik. Így lehetőségünk van arra, hogy a grafikus képernyőt karakteressé tömörítsük. Vagyis ha rajzunk 256 karakternél nem tartalmaz többet, akkor mint szöveget is tárolhatjuk. Ennek fordítottja a kiterítés, amikor a program a tömörített képet a karakterkészlet felhasználásával a grafikus képernyőre másolja. Így azután hatalmas méretű memóriát takaríthatunk meg, ha figyelünk arra, hogy hol van az egyes karaktermezők határa, és ügyelünk rá, hogy a mezőkben csak 256-féle jelet használjunk.

Ebben segít a **Botticelli** másik utasításcsoportja: a kényszerítő hálózat. Ezt bekapcsolhatjuk akár vízszintes, akár függőleges üzemmódba; ilyenkor mindig a megadott üzemmód szerinti iránynak megfelelően, a legközelebbi karakterhatárra kényszeríti a grafikus kurzort.

A másik két programmal kapcsolatban nincs mit hozzáfűznünk az eddigiekhez.

	Morvai	Tóth	Nyisztor	Tallér	Átlag
Botticelli	4,5	4,9	5	5	4,85
TED Paint	4,5	4,2	4,5	4,5	4,18
Paint Box	2,5	2,8	3	3	2,83

ÖSSZESÍTŐ TÁBLÁZAT

Érdemes az előző négy akadály osztályzatait összegezni, s az átlagokat összevetni a szubjektív vélemények osztályzataival

	Alapfunkciók	Különleges szolg.	Kezelhetőség	Tárolás	Átlag
Botticelli	4,58	4,70	4,78	4,85	4,72
TED Paint	4,78	4,48	3,48	4,18	4,22
Paint Box	3,90	2,75	4,08	2,83	3,39

5. AKADÁLY: SZUBJEKTÍV VÉLEMÉNY

A szubjektív véleményekben a futtatóknak nem kellett figyelembe venni előzőleg adott osztályzataikat. Szubjektíve szubjektívek lehettek!

	Morvai	Tóth	Nyisztor	Tallér	Átlag
Botticelli	5	4,8	5	5	4,95
TED Paint	4	4	4	4,5	4,13
Paint Box	4	3,2	3	3	3,3

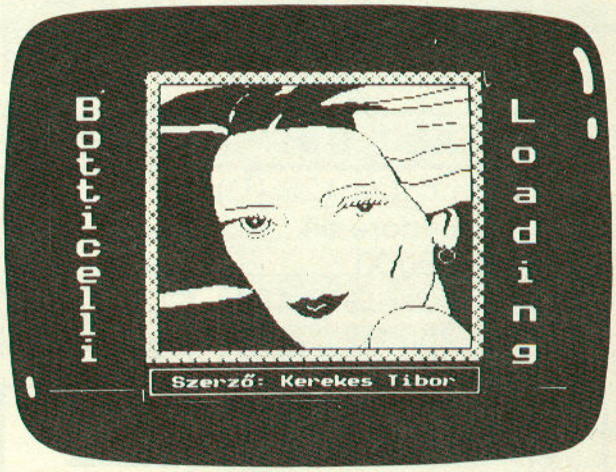
Ismét csak nem fűzünk külön kommentárt az osztályzatokhoz, csupán néhány, a futtatás során „elejtett" véleményt közlünk.

Botticelli

„Csak annyit mondhatok róla, hogy nagyon príma."

„Tökéletes rajzolóprogram lenne – bár ilyen talán nincs is –, ha a koordinátákat is kijelezné munka közben."

„Szépnek szép, jónak is jó – de legalább egyetlen ecsetmintát én magam tudnék megtervezni!"

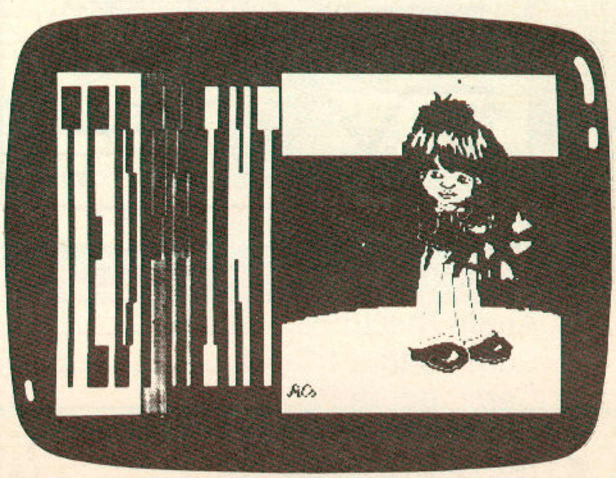


TED Paint

„Ha ilyen soká töltene a Doodle is – pedig nyilván ezt írták át Plus/4-re –, akkor soha nem használtam volna."

„Jól sikerült átirat, egy-két meglepően ügyes új funkcióval."

„Egy magyar készítésű programnál zavar, hogy a menü angolul szól hozzám. Ennek nyilván üzleti okai vannak – de akkor is."



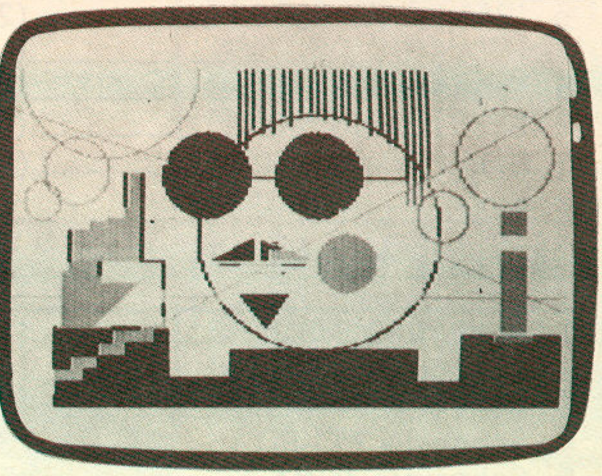
Paint Box

„Aki ezt a programot betette a C–16-os 2,7 szabad KB-jába, annak csak gratulálni tudok. A program ehhez a mérethez képest zseni."

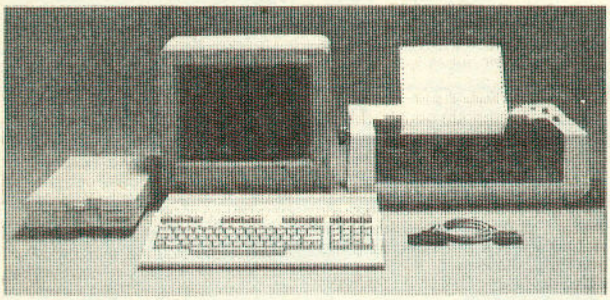
„C–16-on ez a legjobb program, amit mostanáig láttam."

„Nyilván nagyon jól kihasználja a ROM támogatását – de még így is szép programozói teljesítmény ez a színvonal."

„Kevés lehetőséggel kis kerek egészet alkot, nagyon jól használható."

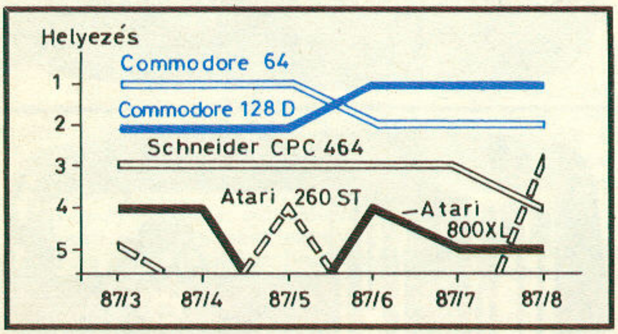


A CHIP című nyugatnémet mikroszámítógépes magazinból. (Az 1987. júliusi, NSZK-beli eladások alapján.) Zárójelben az előző havi helyezés.



HÁZI SZÁMÍTÓGÉPEK

1. Commodore 128 (D) ____ (1)
2. Commodore 64 ____ (2)
3. Atari 800 XL ____ (–)
4. Schneider CPC 464 ____ (3)
5. Atari 260 ST ____ (4)



SZEMÉLYI SZÁMÍTÓGÉPEK

1. Apple Macintosh ____ (1)
2. Commodore PC 10 ____ (6)
3. Commodore PC 20 ____ (5)
4. Commodore Amiga 200 ____ (–)
5. IBM PC/XT 286 ____ (4)
6. IBM PC/AT ____ (2)
7. HP Vectra ____ (–)
8. Tandon PCA ____ (8)
9. Schneider PC–HD ____ (–)
10. IBM PC/XT ____ (3)

FÉLPROFESSZIONÁLIS SZÁMÍTÓGÉPEK

1. Atari 1040 ST ____ (1)
2. Commodore Amiga ____ (4)
3. Atari 520 ST–M ____ (2)
4. Schneider PC–FD ____ (3)
5. Schneider Joyce ____ (5)

ÉKEZETES BETŰK A PLUS/4-ESEN

Ez a kis program azok számára lehet hasznos segédeszköz, akik nem az ÁPISZ-nál vásárolt, ékezetesített géppel rendelkeznek. Futtatása során először a ROM alatti RAM-területre másolja a karaktergenerátort, és azt úgy módosítja, hogy az megfeleljen az ékezetes gépek ROM karakterkészletének. Kisbetűs üzemmódban a C= billentyű egyidejű lenyomásával a következő billentyűkkel érhetjük el az ékezetes betűket:

Q	**W**	**E**	**R**	**T**	**Z**	**U**	**I**	**O**
Á	Ü	É	Ű	Ő	Ö	Ú	Í	Ó
A	**S**	**D**	**F**	**G**	**H**	**J**	**K**	**L**
á	ü	é	ű	ő	ö	ú	í	ó

Ezt a karakterkészletet „bármilyen" BASIC programhoz használhatjuk. Aktivizálása a **POKE 65298,192** utasítással történik (lásd a 420-as sort). POKE 2042,0 kiadásával az osztott (GRAPHIC2; GRAPHIC4) grafikus képernyő szöveges ablakában is használhatjuk az általunk készített karaktereket. Ha ezután GRAPHIC0-lát adunk, a karakterkészlet eltűnik. Ezért tettük fel a program 410-es sorában a POKE 65298,192-őt a HELP billentyűre is. Így ilyenkor ennek lenyomása segít.

Az az igazság, hogy a program írása közben sokat kínlódtam az ékezetes karakterek osztott képernyőn való megjelenítésével. Végül is a rejtély megoldásában Morvai László segített. Hogy mitől rejtély a rejtély – ennek megírására őt kértem meg.

Meszlényi Zoltán

```
1 REM ********************************
2 REM *   C= UJSAG   SORSZAM:   073   *
3 REM *   EKEZETES BETUK             *
4 REM *   PROGRAM: MESZLENYI ZOLTAN   *
5 REM ********************************
6 REM
10 POKE56,207:CLR
20 PRINT"▒▒▒▒▒▒▒TD000 D800 D000▒"
30 FOR I=1319 TO 1326:READJ:POKEI,J:NEXT
40 POKE239,8
50 MONITOR
60 PRINT"▒▒"+CHR$(14)
100 DATA13,88,13,71,207,54,48,13
110 DATA0C,18,00,38,18,18,3C,00
120 DATA0C,18,3C,18,18,18,3C,00
130 DATA33,66,3C,66,66,66,3C,00
140 DATA00,00,00,00,00,00,00,FF
150 DATA33,66,00,3C,66,66,3C,00
160 DATACC,CC,33,33,CC,CC,33,33
170 DATA01,01,01,01,01,01,01,01
180 DATA00,00,00,00,CC,CC,33,33
190 DATACC,99,33,66,CC,99,33,66
200 DATA03,03,03,03,03,03,03,03
210 DATA0C,18,3C,66,7E,66,66,00
220 DATA0C,18,3C,66,7E,60,3C,00
230 DATA18,18,18,1F,1F,00,00,00
240 DATA00,66,00,66,66,66,3E,00
250 DATA00,00,00,00,00,00,FF,FF
260 DATA0C,18,3C,06,3E,66,3E,00
270 DATA0C,18,7E,60,78,60,7E,00
280 DATA33,66,00,66,66,66,3C,00
290 DATA66,00,66,66,66,66,3C,00
300 DATA00,66,00,3C,66,66,3C,00
310 DATA0C,18,00,66,66,66,3E,00
320 DATA0C,18,00,3C,66,66,3C,00
330 DATA66,00,3C,66,66,66,3C,00
340 DATA0C,18,66,66,66,66,3C,00
350 DATA0C,18,3C,66,66,66,3C,00
360 DATA01,03,06,6C,78,70,60,00
370 DATA33,66,00,66,66,66,3E,00
380 FOR I=55048 TO 55263
390 READ D$:POKE I,DEC(D$)
400 NEXT
410 KEY8,"POKE65298,192"+CHR$(13)
420 POKE65298,192
430 PRINT"  EKEZETES KARAKTERKESZLET";
440 PRINT" $D000-$D800"
```

Újabb ördöngösségek a PLUS/4-en

Aki olyan programot próbált már írni, amelyben saját, szerkesztett karakterkészletét és valamelyik grafikus üzemmódot szerette volna együtt használni, valószínűleg sokat mérgelődött.

A GRAPHIC 0 utasítás és az osztott képernyős üzemmódok – ha tetszik, ha nem –, visszaállítják a $FF12 mutatót, így a gép a karakterkészletet nem a RAM-ban, hanem a ROM-ban „keresi", ennek pedig a legtöbb esetben zavaros képernyő az eredménye.

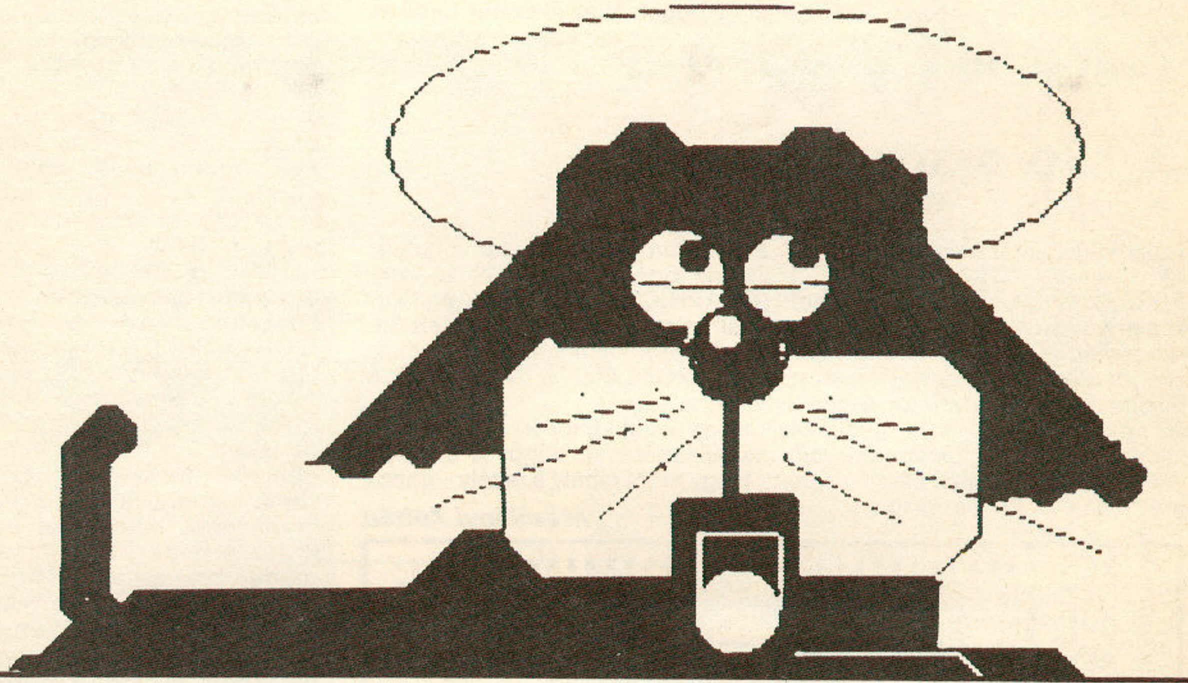
A GRAPHIC 0 hatását viszonylag könnyű kivédeni, elég, ha egy POKE utasítással visszaállítjuk a karakterkészletet. Elegánsabb megoldás, ha saját gépi kódú szubrutint készítünk, amely anélkül válthat karakteres kijelzésre, hogy a karaktergenerátor helyzete megváltozna. A GRAPHIC 0 utasítás a gép video reset rutinját hívja meg, amely a $c7c9–c7ef tartományban helyezkedik el. Ezt a rutint kell módosítanunk olyanformán, hogy kihagyjuk belőle a $c7e3–$c7e8 közé eső részt. A módosított szubrutin így néz ki:

```
lda  $ff06
and #$df
sta  $ff06
lda  $ff07
and #$ef
sta  $ff07
lda  $ff14
and #$07
ora  #$08
sta  $ff14
lda #$00
sta  $83
rts
```

Az osztott képernyő okozta galibáért az interrupt-ciklus a felelős. Osztott képernyős üzemmódban először logikai vagy-műveletet végez a $ff12 és a $07fa címek között, majd az eredményt visszatölti az $ff12 címre. A $07fa cím tartalma alapállapotban $04, ezt a RAM inicializáló rutin ($f352–$f3cf) tölti be ide, amely csak bekapcsoláskor és minden RESET alkalmával aktivizálódik. Ha a programban kiadunk egy POKE DEC("07FA"),0 azaz POKE 2042,0 utasítást, a továbbiakban már használhatjuk a saját karakterkészletünket az osztott grafikával együtt is.

Morvai László

PLUTÓ rajzoló program



Ez a program természetesen nem veheti fel a versenyt a profi rajzoló programokkal, de nem is erre vállalkozik. Csupán a plusz 4-es BASIC grafikus utasításait teszi könnyen hozzáférhetővé. Mindent kijelölhetünk a képernyőn joystickkel és a megfelelő billentyűkkel. Ezen kívül néhány olyan szolgáltatást nyújt programunk, amiről egy rajzolóprogram nem mondhat le. Lehetőséget ad az elkészült rajzocskák kinyomtatására, kazettán vagy lemezen történő tárolására. Így hát egyszerűbb feladatok megoldására egészen jól használható, és a programlista is segítséget nyújthat, ha valakinek gondot okoz egy-egy grafikus utasítás alkalmazása.

A program elindítás után egy HELP képernyővel jelentkezik be. Ha az itt található utasításokat elolvastuk, a következő HELP képernyőre a szóközbillentyűvel léphetünk. Az ESC billentyűvel léphetünk ki a HELP-ből, ezután kezdhetjük el a rajzolást. Rajzolás közben a HELP billentyűvel bármikor megnézhetjük a HELP képernyőket.

A rajzlap közepén megjelenő „célkereszt" jelzi ceruzánk helyzetét. Minden további utasítás erre a helyre vonatkozik.

LEHETŐSÉGEK

A CERUZA MOZGATÁSA

A kis keresztet az 1. portba kapcsolt joystickkel nyolc irányba, képpontonként mozgathatjuk, állandó sebességgel. Ha közben lenyomva tartjuk a **tűzgombot,** vonalat tudunk rajzolni. Így, képpontonként lépegetve, bizony nem valami gyorsan jutna el a ceruza a képernyő egyik széléről a másikra. A **kurzorbillentyűk** segítségével mind a négy irányba nyolc pontos ugrásokkal mozoghatunk. A ceruzát ki is lehet vinni a képernyőről, erre pl. nagy sugarú körívek rajzolásánál lehet szükségünk. A **HOME** billentyűvel bármikor visszavihetjük a ceruzát a képernyő közepére. A ceruza aktuális helyzetét meg is tudjuk jegyezni. Egyidejűleg négy pontot jelölhetünk meg a C= **1...**C= **4** billentyűkkel. Az így megjelölt pontokra az **1...4** billentyűkkel vihetjük vissza a keresztet.

VONAL RAJZOLÁSA

Vigyük a ceruzát oda, ahol kezdeni akarjuk a vonalat, nyomjuk meg a V billentyűt, erre a kis kereszt villogni kezd. Most a villogó keresztet tudjuk mozgatni a joystickkel vagy a billentyűkkel. A villogó kereszt helyén egy álló kereszt marad, ez jelzi a rajzolás kiinduló helyzetét. Ha lenyomjuk a **tűzgombot** vagy a **szóközbillentyűt,** kirajzolódik a két kereszt által kijelölt egyenes. Ceruzánk hegye a vonal végpontjába kerül (a villogó kereszt helyére). Ha a munkát a vonal kiinduló pontjából akarjuk folytatni, nyomjuk meg a £ billentyűt.

KÖR

Az egyeneshez hasonló a körrajzoló funkció működtetése. Vigyük a ceruzát a kör középpontjába. Nyomjuk meg a K billentyűt, a villogó keresztet mozgassuk a körvonal tetszőleges pontjára és nyomjuk meg a **tűzgombot,** vagy a **szóközbillentyűt.**

ELLIPSZIS

Az E billentyűvel tudjuk ezt a funkciót aktivizálni. Az álló kereszt az ellipszis középpontját (a kistengely és a nagytengely metszéspontját), a villogó kereszt pedig a két sugár (vagyis a fél nagytengely és a fél kistengely) által meghatározott téglalap csúcspontját jelöli ki.

NÉGYSZÖG

Az N billentyűvel rajzolhatunk derékszögű négyszögeket (téglalapot, négyzetet). A két kereszt a négyszög két szemben levő csúcspontját határozza meg. A **SHIFT**+**N** billentyűvel beszínezett négyszöget tudunk rajzolni.

SZAKASZ FELEZÉS

Az = billentyű lenyomására ceruzánk az álló és a villogó célkereszt által kijelölt szakasz felezőpontjába kerül.

FESTÉS

Az **F** billentyűvel befesthetjük azt a zárt területet, amelynek belsejében van a ceruza. Vigyázzunk ezzel a funkcióval, mert ha a terület nem teljesen zárt, a festék bizony kifolyik, és könnyen befesthetjük az egész rajzlapot! A festést meg tudjuk állítani a **STOP** billentyűvel. Többszínű üzemmódban lehet hasznos a **SHIFT+F** billentyű, ezzel úgy lehet mázolni, hogy bármilyen nem háttérszínű határoló vonalnál befejeződik a terület feltöltése.

BETŰK

A **B** billentyű lenyomása után a ceruza által meghatározott karakterpozíciótól kezdve szöveget írhatunk a képre (ékezetes betűket is!). A szöveget **RETURN**-nel vagy az **ESC**-vel zárjuk le, ceruzánkat a kiinduló pontban kapjuk vissza. **SHIFT+B**-vel inverz betűket írhatunk.

EGYSZÍNŰ, TÖBBSZÍNŰ

Alaphelyzetben a program normál nagyfelbontású (HI-RES) grafikus üzemmódban dolgozik. Így egyszerre (egy 8×8-as karakterpozíción belül) csak két színt használhatunk, az egyik az a szín, amivel rajzolunk, a másik a háttér színe. A többszínű (multicolor) grafikát az **F2** billentyűvel kapcsolhatjuk be. Ebben az üzemmódban négy különböző színforrást használhatunk, de a vízszintes képfelbontás a felére csökken, azaz minden képpont kétszer olyan széles lesz. Multicolor üzemmódban a célkereszt is kétszer olyan széles, mint HI-RES-ben. **F1**-gyel visszatérhetünk HI-RES-be.

FIGYELEM! A grafikus üzemmódváltás törli a képernyőt!

SZÍNEK

Multicolorban négy színforrásból választhatjuk ki a **CTRL+1...CTRL+4** billentyűkkel, hogy ceruzánknak milyen színű hegye legyen. HI-RES-ben ezek közül csak a **CTRL+1** és a **CTRL+4** aktív. A színforrásokhoz tartozó tényleges színeket egy kis menüből jelölhetjük ki. Ha lenyomjuk a * billentyűt, a képernyő alsó öt sorában egy ablak nyílik, itt tudjuk beállítani a CTRL+1...CTRL+4 billentyűk folyamatos lenyomásával a színeket. Az éppen kiválasztott színforrást egy > jel jelzi. A színválasztó menüből **ESC**-vel léphetünk ki.

TÖRLÉS

A teljes képernyőt letörölhetjük a **CLEAR** billentyűvel. Egyes részletek javításához használható a radír. Ez egy kis téglalap, amely a **DEL** billentyűvel kapcsolható be a ceruza helyén, és ugyanúgy lehet mozgatni, mint a ceruzát. Legyünk óvatosak a radírral, mert mindent letöröl, ami az útjába kerül! Az **ESC** billentyű lenyomására visszakapjuk a radír helyén a ceruzát. Finomabb (képpont méretű) javításhoz a ceruzát is használhatjuk, háttérszínű heggyel.

MEMÓRIA

Az **M** billentyűvel léphetünk be a memóriakezelő menübe, ahonnan két egymástól független memóriába tudjuk rajzunkat eltenni, vagy onnan visszahívni.

BLOKKOK

Az **S** billentyű lenyomása után kijelölhetünk egy téglalap alakú területet (úgy mintha négyszöget rajzolnánk). Ezt a blokkot a program megjegyzi, és a rajzlapon egy billentyű lenyomására bárhol meg tudja jeleníteni a következő öt utasítás valamelyikével:

G: Változtatás nélküli megjelenítést eredményez. Vagyis az egész téglalappal felülírja az alatta levő rajzot.

I: Inverz megjelenítés. Úgy jeleníti meg a blokkot, hogy a téglalapon belül a háttér- és tintaszínű pontok felcserélődnek.

O: A blokk és a háttér pontonkénti VAGY kapcsolata jelenik meg, azaz a képen ott lesznek pontok, ahol a

```
1 REM *********************************
2 REM *   C= UJSAG   SORSZAM: 074      *
3 REM *   PLUTO                        *
4 REM *   PROGRAM: MESZLENYI ZOLTAN    *
5 REM *********************************
6 REM
10 REM  BASIC MUNKATERULET $4000-$8000
15 :
20 GRAPHIC 1,1:POKE 56,127:CLR
25 :
30 GOSUB 1590:REM INICIALIZALAS
35 GOSUB 1185:REM HELP SCREEN
40 :
45 REM GRAPHIC 1  GRAPHIC 2 BELEPES
50 :
55 G=1:Q3$="▚▌":GOTO 65
60 G=2:Q3$="▚▌▐▐":REM CTRL+4123
65 GRAPHIC G*2-1,1
70 :
75 REM CERUZA  RADIR
80 :
85 DRAW,4,. TO 4,1:DRAW,4,5 TO 4,6
90 DRAW,.,3 TO 1,3:DRAW,7,3 TO 8,3
95 SSHAPE Q$(4),.,.,8,6:SCNCLR
100 BOX,1,1,7,5
105 SSHAPE Q$(.),.,.,8,6:SCNCLR
110 X=160/G:Y=100
115 :
120 REM  FOPROGRAM
125 :
130 GRAPHIC G*2-1
135 :
140 TRAP 10010:REM  CSAK TESZTELES UTAN!
145 GSHAPE Q$(W),X-4,Y-3,W
150 DO
155 :GET K$
160 :DO UNTIL JOY(1)=.
165 ::J=JOY(1) AND 127
170 ::GSHAPE Q$(W),X-4,Y-3,W
175 ::X=X+X(J):Y=Y+Y(J)
180 ::IF JOY(1)>127 THEN DRAWS,X,Y
185 ::GSHAPE Q$(W),X-4,Y-3,W
190 ::IFW=. THEN POKE 239,.
195 :LOOP
200 LOOP WHILE K$=""
205 GSHAPE Q$(W),X-4,Y-3,4
210 GOSUB 465
215 IF K$="F" THEN PAINT S,X,Y
220 IF K$=CHR$(198) THEN PAINT S,X,Y,1
225 K=INSTR(K1$,K$)
230 IF K>10 THEN GOSUB 390
235 ON K GOSUB 355,355,355,355,355,580,5
75,265,365,370,315,320,320,325,330,335,3
40
240 K=INSTR(K2$,K$)+1
245 ON K GOTO 140,855,65,55,60
250 :
255 REM  KILEPES
260 :
265 GRAPHIC G*2:POKE 2022,20
270 PRINT"▣      KILEPES A PROGRAMBOL:▣ES
C▣":PRINT
275 PRINT"          FOLYTATAS:▣SPACE▣"
280 GETKEY VZ$
285 IF VZ$=ES$ THEN END
290 PRINT ES$"N":GRAPHIC G*2-1
295 RETURN
300 :
305 REM  RAJZOLO RUTINOK
310 :
315 DRAW S,V,Z TO X,Y:RETURN
320 BOX S,V,Z,X,Y,,K-12:RETURN
325 CIRCLE S,V,Z,ABS(V-X),ABS(Z-Y):RETUR
N
330 CIRCLE S,V,Z,SQR((V-X)↑2+((Z-Y)/G)↑2
):RETURN
335 X=V+INT((X-V)/2):Y=Z+INT((Y-Z)/2):RE
TURN
340 L=INT((ABS(V-X)+1)/8/G+.99)*(ABS(Z-Y
)+1)+4
345 IF L>255 THEN GOSUB 535:RETURN
350 SSHAPE S$(C),V,Z,X,Y:RETURN
355 GSHAPE S$(C),X,Y,K-1:RETURN
360 :
365 W=.:RETURN
370 W=4:RETURN
375 :
380 REM  VONAL, KOR, NEGYSZOG KIJELOLES
385 :
390 V=X:Z=Y
395 DO UNTIL JOY(1)=128
400 :J=JOY(1).AND 127
405 :GSHAPE Q$(W),X-4,Y-3,4
410 :X=X+X(J):Y=Y+Y(J)
415 :GET K$:GOSUB 465
420 :IF K$=ES$ THEN K=.:GOSUB 535
425 :GSHAPE Q$(W),X-4,Y-3,4
430 LOOP UNTIL K$=" " OR K$=ES$
435 GSHAPE Q$(W),V-4,Z-3,4
440 GSHAPE Q$(W),X-4,Y-3,4
445 RETURN
450 :
455 REM  CERUZA MOZGATAS
460 :
465 Q=INSTR(Q4$,K$)*2-1
470 IF Q>. THEN X=X+X(Q)*8:Y=Y+Y(Q)*8:RE
TURN
475 Q=INSTR(Q2$,K$)
480 IF Q THEN V(Q)=X:Z(Q)=Y:RETURN
485 Q=INSTR(Q3$,K$)
490 IF Q THEN S=Q-1:RETURN
495 Q=INSTR(Q5$,K$)
500 IF Q THEN X=V(Q):Y=Z(Q):RETURN
505 Q=INSTR(Q1$,K$)
510 IF Q THENC=Q:RETURN
515 Q=INSTR(Q6$,K$)
520 ON Q GOSUB 530,535,1015,660,1185,555
525 RETURN
530 X=160/G:Y=100:RETURN
535 X=V:Y=Z:RETURN
540 :
545 REM  HARDCOPY RUTIN HIVAS
550 :
555 SYS 4160:RETURN
560 :
565 REM  BETUK
570 :
575 RV=1
580 A=X/8*G:B=Y/8
585 POKE 740,20
590 REM  ROM KARAKTERGENERATOR ESETEN
595 REM  POKE 740,212!
600 DO
605 :GETKEY B$
610 :IF B$=RT$ OR B$=ES$ THEN EXIT
615 :IF B$=CHR$(140) THEN GOSUB 1185:GOT
O 630
620 :CHAR S,A,B,B$,RV
625 :A=A+1:IF A>39 THEN A=.:B=B+1
630 LOOP UNTIL B>25
635 RV=0
640 RETURN
645 :
```

kijelölt téglalapban VAGY a háttérben van tintaszínű képpont.
A: A blokk és a háttér pontonkénti ÉS kapcsolatát jeleníti meg. Csak oda rajzol pontot, ahol a blokkban is, ÉS a háttérben is tintaszínű a képpont.

X: Blokk és háttér kizáró VAGY (XOR) kapcsolata. Azokat a pontokat rajzolja meg, amelyek vagy csak a blokkban, vagy csak a háttérben tintaszínűek. Az X többszöri ismétlésével azt is ki tudjuk próbálni, hogyan nézne ki a rajz, ha a blokkot az adott helyre másolnánk.

```
650 REM  MEMORIA
655 :
660 POKE 2022,20:GRAPHIC G*2:PRINT"▢"
665 PRINT"  ▒ 1 ▒:I.MEMORIABA   ▒ 2 ▒:II
.MEMORIABA":PRINT
670 PRINT"  ▒ 3 ▒:I.MEMORIABOL  ▒ 4 ▒:II
.MEMORIABOL"
675 GETKEY A$
680 GRAPHIC G*2-1
685 IF A$=ES$ THEN RETURN
690 A=INSTR(Q5$,A$)+1
695 ON A GOTO 660,710,745,780,815
700 :
705 REM  $1800-$3FFF => $A800-$CFFF
710 :POKE 34,0:POKE 35,208
715 :POKE 36,0:POKE 37,64
720 :POKE 2035,0:POKE 2036,40
725 :SYS 51192
730 RETURN
735 :
740 REM  $1800-$3FFF => $8000-$A7FF
745 :POKE 34,0:POKE 35,168
750 :POKE 36,0:POKE 37,64
755 :POKE 2035,0:POKE 2036,40
760 :SYS 51192
765 RETURN
770 :
775 REM  $A800-$CFFF => $1800-$3FFF
780 :POKE 34,0:POKE 35,64
785 :POKE 36,0:POKE 37,208
790 :POKE 2035,0:POKE 2036,40
795 :SYS 51192
800 RETURN
805 :
810 REM  $8000-$A7FF => $1800-$3FFF
815 :POKE 34,0:POKE 35,64
820 :POKE 36,0:POKE 37,168
825 :POKE 2035,0:POKE 2036,40
830 :SYS 51192
835 RETURN
840 :
845 REM  LOAD-SAVE
850 :
855 POKE 2022,20:GRAPHIC G*2
860 PRINT"▢"
865 PRINT"  ▒ 1 ▒:"U$"RA   ▒ 2 ▒:"U$"ROL
":PRINT
870 IF U=8 THEN PRINT"  ▒ 3 ▒:DIRECTORY"
:
875 PRINT"  ▒ESC▒:KILEPES";
880 GETKEY A$
885 IF A$=ES$ THEN 130
890 IF U=8 AND A$="3" THEN PRINT ES$"N":
GRAPHIC 0:DIRECTORY:GETKEY A$:GOTO 855
895 IF A$<"1" OR A$>"2" THEN 855
900 PRINT ES$"D   RAJZ NEVE: ";
905 PRINT ES$"T"SPC(16)ES$"B▒";
910 INPUT N$:PRINT ES$"N"
915 POKE 2022,14:PRINT"▒▒▒▒▒▒▒";
920 :
925 REM  MONITOR UTASITAS KEPERNYORE
930 :
935 IF A$="1" THEN PRINT"S"CHR$(34)N$CHR
$(34)U"1800 4000▒"
940 IF A$="2" THEN PRINT"L"CHR$(34)N$CHR
$(34)U"▒"
945 :
950 REM  BILLENTYUPUFFERBE
955 :
960 RESTORE 995
965 FOR I=1319 TO 1327
970 :READ A:POKE I,A
975 NEXT
980 :
985 REM RETURN X RETURN GOTO 130 RETURN
990 :
995 DATA 13,88,13,71,207,49,51,48,13
1000 POKE 239,9
1005 MONITOR
1010 :
1015 REM  SZIN
1020 :
1025 POKE 2022,20:PRINT"▢"
1030 FOR I=. TO 4
1035 :C(I)=RCLR(I)
1040 :IF C(I)=. THEN C(I)=16
1045 :L(I)=RLUM(I)
1050 NEXT
1055 GRAPHIC G*2
1060 DO
1065 :IF RCLR(0)=1 THEN W$=CHR$(5):ELSE
W$=CHR$(144)
1070 :PRINT"▒"W$;
1075 :FOR J=1 TO 3
1080 ::IF J=S THEN PRINT TAB(6)W$">";
1085 ::PRINT TAB(8)W$;"CTRL";J;": COLOR"
;
1090 ::PRINT USING"## ";J,C(J),L(J);
1095 ::COLOR 1,C(J),L(J)
1100 ::PRINT"▒        ▒"
1105 :NEXT
1110 :IF S=. THEN PRINT TAB(6)W$">";
1115 :PRINT TAB(8)W$"CTRL 4 : COLOR 0 ";
1120 :PRINT USING"## ";C(0),L(0)
1125 :GETKEY A$
1130 :I=INSTR(Q3$,A$)-1
1135 :IF I<0 OR I>4 THEN EXIT
1140 :IF L(I)<7 THEN L(I)=L(I)+1:ELSE L(
I)=0:IF C(I)<16 THEN C(I)=C(I)+1:ELSE C(
I)=1
1145 :COLOR I,C(I),L(I)
1150 LOOP
1155 COLOR1,C(1),L(1):PRINT"▒▒"
1160 GRAPHIC G*2-1
1165 RETURN
1170 :
1175 REM  HELP
1180 :
1185 GRAPHIC 0
1190 :
1195 POKE 65298,192:REM  CSAK RAM
1200 REM  KARAKTERGENERATOR ESETEN
1205 REM  KELL BEIRNI!
1210 :
1215 C0=RCLR(0):IF C0=. THEN C0=16
1220 C1=RCLR(1):IF C1=. THEN C1=16
1225 L0=RLUM(0):L1=RLUM(1)
1230 COLOR 0,1:COLOR 1,2
1235 PRINT ES$"N"
1240 CHAR, 1,0,"▒HELP▒:UTASITASOK"
1245 CHAR,17,0,"▒ * ▒:SZINEK"
1250 CHAR, 1,2,"▒ F1 ▒:NORMAL"
1255 CHAR,17,2,"▒ F2 ▒:MULTICOLOR"
1260 CHAR, 1,4,"▒HOME▒:KOZEPRE"
1265 CHAR,17,4,"▒CLEAR▒:KEPERNYO TORLES"
1270 CHAR, 1,6,"▒DEL▒:RADIR"
1275 CHAR,17,6,"▒ £ ▒:ELOZO HELYRE"
1280 CHAR, 1,8,"▒ V ▒:VONAL"
1285 CHAR,17,8,"▒ = ▒:SZAKASZ FELEZES"
1290 CHAR, 1,10,"▒ K ▒:KOR"
1295 CHAR,17,10,"▒ E ▒:ELLIPSZIS"
1300 CHAR, 1,12,"▒ B ▒:BETUK"
1305 CHAR,17,12,"▒SH+B▒:INVERZ BETUK"
1310 CHAR, 1,14,"▒ N ▒:NEGYSZOG"
1315 CHAR,17,14,"▒SH+N▒:NEGYSZOG FESTVE"
1320 CHAR, 1,16,"▒ F ▒:FEST"
1325 CHAR,17,16,"▒SH+F▒:MAZOL"
1330 CHAR, 1,18,"▒ M ▒:MEMORIA"
1335 CHAR,17,18,"▒ @ ▒:LOAD, SAVE"
1340 CHAR, 1,20,"▒CTRL+P▒:PRINT"
1345 CHAR,17,20,"▒CTRL+V▒:PROGRAM VEGE"
1350 CHAR,10,22,"KOVETKEZO LAP:SPACE"
1355 CHAR,15,24,"RAJZOLAS:ESC"
1360 :
1365 GETKEY Y$:IF Y$=ES$ THEN 1445
1370 :
1375 CHAR,1,0,"▢  ▒ S ▒:BLOKK KIJELOLES"
1380 CHAR,1,2,"BLOKK MEGJELENITES:"
1385 CHAR,2,4,"▒ G ▒:VALTOZTATAS NELKUL"
1390 CHAR,2,6,"▒ I ▒:INVERZBEN"
1395 CHAR,2,8,"▒ O ▒:BLOKK  OR HATTER"
1400 CHAR,2,10,"▒ A ▒:BLOKK AND HATTER"
1405 CHAR,2,12,"▒ X ▒:BLOKK XOR HATTER"
1410 CHAR,1,14,"▒CTRL+A▒-▒H▒:BLOKK NEVE"
1415 CHAR,1,16,"▒CTRL+1▒-▒4▒:SZINFORRAS"
1420 CHAR,1,18,"▒C=1▒-▒4▒:PONT JELOLES"
1425 CHAR,1,20,"▒1▒-▒4▒:JELOLT PONTRA UG
RAS"
```

Egyszerre nyolc különböző blokkal dolgozhatunk. A blokk-kijelölés előtt a **CTRL + A...CTRL + H** billentyűkkel nevezhetjük meg az aktuális blokkot (azaz hogy éppen melyiket akarjuk írni vagy olvasni). Alaphelyzetben a CTRL + A blokk aktív. Egy blokkra mindaddig emlékszik a program, míg azt újabb kijelöléssel felülírjuk. Egy-egy blokk mérete korlátozott (max. 255 byte). Ha a kijelölés során ezt a korlátot túllépjük, a kijelölést a program nem veszi figyelembe. Ebben az esetben a ceruzát az álló kereszt helyén kapjuk vissza. Ha a kijelölt terület elfér egy blokkban, a kijelölés után a ceruza a villogó kereszt helyére kerül.

```
1430 CHAR,1,22,"■ESC■: KILEPES (RADIR,BE
TUK,SZIN)"
1435 CHAR,4,24,"ELOZO LAP:SPACE  RAJZOLA
S:ESC"
1440 GETKEY Y$:IF Y$<>ES$ THEN 1235
1445 COLOR 1,C1,L1:COLOR 0,C0,L0
1450 GRAPHIC G*2-1
1455 RETURN
1460 :
1465 REM  HARDCOPY RUTIN $1040-$10FE
1470 :
1475 DATA A9,60,85,AD,A9,04,85,AE,20,56
1480 DATA E1,A5,AD,20,93,FF,A9,08,20,1D
1485 DATA E2,A2,08,A9,00,95,D0,CA,10,F9
1490 DATA A9,00,8D,AD,02,8D,AE,02,A9,00
1495 DATA 85,D0,A5,D2,18,65,D0,8D,AF,02
1500 DATA 20,06,C0,18,A5,61,F0,01,38,66
1505 DATA D3,E6,D0,A6,D0,E0,07,D0,E5,EE
1510 DATA AD,02,D0,03,EE,AE,02,A5,D3,38
1515 DATA 6A,20,BD,10,AD,AD,02,C9,40,D0
1520 DATA CB,AD,AE,02,F0,C6,A9,0D,20,F0
1525 DATA 10,A5,D2,18,69,07,85,D2,C9,C9
1530 DATA 90,AE,A9,0D,20,1D,E2,A9,0F,20
1535 DATA 1D,E2,4C,3D,E2,C5,D1,D0,1E,E6
1540 DATA D8,D0,19,C6,D8,A9,1A,20,1D,E2
1545 DATA A5,D8,20,1D,E2,A5,D1,20,1D,E2
1550 DATA A9,01,85,D8,A5,D6,85,D1,60,85
1555 DATA D6,A6,D8,F0,F1,E0,03,B0,DE,A5
1560 DATA D1,CA,F0,E5,D0,DE,A6,D1,E0,80
1565 DATA D0,C7,A2,00,86,D6,F0,D7,EA,21
1570 DATA 40
1575 :
1580 REM  INICIALIZALAS
1585 :
1590 RESTORE 1475
1595 FOR I=1 TO 191
1600 :READ A$
1605 :POKE 4159+I,DEC(A$)
1610 :P=P+DEC(A$)
1615 NEXT
1620 IF P=26332 THEN 1660
1625 PRINT" ■ADATHIBA!":GRAPHIC 0:END
1630 :
1635 REM CHAR UTASITASHOZ
1640 REM $D400-$D7FF => $1400-$17FF
1645 REM CSAK RAM KARAKTERGENERATOR
1650 REM ESETEN KELL BEIRNI 1660-1675-IG
1655 :
1660 :POKE 34,0:POKE 35,24
1665 :POKE 36,0:POKE 37,216
1670 :POKE 2035,0:POKE 2036,4
1675 :SYS 51192:POKE 2042,0
1680 :
1685 COLOR 4,1
1690 FOR I=1 TO 8
1695 :KEY I,CHR$(132+I)
1700 :Q1$=Q1$+CHR$(I)
1705 :READ X(I),Y(I)
1710 NEXT
1715 DATA.,-1,1,-1,1,.,1,1
1720 DATA.,1,-1,1,-1,.,-1,-1
1725 G=1:S=1:C=1:W=4
1730 RT$=CHR$(13):ES$=CHR$(27)
1735 K1$="GIOAXB"+CHR$(194)+CHR$(22)+CHR
$(20)+ES$+"VN"+CHR$(206)+"EK=S"
1740 K2$="@■"+CHR$(133)+CHR$(134)
1745 Q2$="■■■■":REM  C=1234
1750 Q4$="■■■■":REM  FEL JOBB LE BAL
1755 Q5$="1234"
1760 Q6$="■£*M"+CHR$(140)+CHR$(16)
1765 U=PEEK(174)
1770 U$="FLOPPY":IF U<8 THEN U$=" MAGNO"
1775 RETURN
1780 :
10000 REM STOP BILLENTYU HIBACSAPDA
10005 :
10010 RESUME NEXT
```

TÁROLÁS

Ezt a kis menüt a @ billentyűvel választhatjuk ki, innen a beépített **MONITOR L** és **S** parancsát aktivizálva az elkészült rajzot elmenthetjük ill. visszatölthetjük. A tárolás azon a perifériaegységen történik, amelyről a programot betöltöttük. Ha drive-ot használunk, a lemez tartalomjegyzékét is megnézhetjük.

HARDCOPY

A **CTRL + P** billentyűvel egy egyszerű kis hardcopy rutint hívhatunk meg, segítségével rajzunkat MPS 803-as (vagy ezzel kompatíbilis) printeren kinyomtathatjuk.

KILÉPÉS A PROGRAMBÓL

CTRL + V billentyű. Kilépési szándékunkat még az ESC billentyűvel meg kell erősíteni, nehogy szándékunk ellenére fejezzük be a rajzolást.

A PROGRAM BEGÉPELÉSE, FUTTATÁSA

A programlistát célszerűen ékezetes betűkkel gépeljük be, hogy a program üzenetei jól olvasható formában jelenjenek meg a képernyőn. Mivel vannak **ékezetes** karakterkészlettel rendelkező gépek és **ékezetes** betűk **nélküliek**, a program begépelése, futtatása némileg különbözik a kétféle gépen.

ÉKEZET NÉLKÜLI GÉPEK

Segítségül kell hívni a 25. oldalon található 07. sorszámú programot. A rajzoló beírása és minden futtatása előtt be kell írni, illetve tölteni, majd lefuttatni az ékezetes karakterkészletet előállító programot. A rajzoló **585, 1195, 1660–1675** számú sorainak utasításai biztosítják azt, hogy a definiált karakterkészlet a grafikus képernyőn, illetve grafikus üzemmód mellett a szöveges képernyőn is megjeleníthető legyen.

ÉKEZETES GÉPEKEN

Az 1195-ös és az 1660–1665–1670–1675-ös sorokat hagyjuk ki. Az 585-ös sorba ezt írjuk: POKE 740,212. Ezekre a változtatásokra a lista megjegyzései is felhívják a figyelmet!

TESZT

A 140-es sorba a TRAP utasítást **csak akkor** írjuk be, ha már teszteltük a programot, és minden funkció megfelelően működik. Ezek után a stop billentyűvel leállíthatjuk a PAINT utasítás végrehajtását, de a TRAP rutin hibaüzenet nélkül átugraná a ki nem javított gépelési hibákat is.
A 935–940-es sorok egy szabályos MONITOR utasítást írnak ki a képernyőre. Ezt látjuk is. Az utasítás megfelelő megváltoztatásával turbót is használhatunk a kazettás tároláshoz.

TOVÁBB!

A MONITOR parancsok végrehajtása a billentyűpufferbe POKE-olt karakterkódokkal történik, ez biztosítja a MONITOR-ból kilépést, és a BASIC program folytatását is GOTO 130-cal. Ezt is megfelelően módosítani kell, ha a programot átsorszámozzuk!
Az inicializáló rész (1580–1775) tölti fel a K1$, K2$, Q1$...Q6$ változókat a vezérlő billentyűk ASCII kódjával. Ezek átírásával, és a hozzájuk tartozó ON... GOSUB... utasítások módosításával programunk tetszőleges rutinokkal bővíthető. Pl.: elforgatást, körív, sokszög rajzolást is megvalósíthatunk, a rajzoló programot saját igényeinknek megfelelően alakíthatjuk.

Meszlényi Zoltán

Kérem szíveskedjenek az alábbi a C 16 és C 64 számítógépre vonatkozó kérdéseimre választ adni.
1. A fent említett számítógépek felhasználható memóriaterülete hány kilobyte?
2. A felhasználható memóriaterület hogyan és hány kilobyte-ra növelhető?
3. Mennyibe kerül a tárbővítés?
***Kőhalmy József,* 1071. Bp., Damjanich u. 49.**

A C 16 RAM-területe 16, a C 64-é pedig 64 kilobyte. Mindkettőnek egy részét elfoglalják azonban a rendszertáblák, továbbá a képernyőmemória és a szín-mátrix (ez a C 16-nál összesen kb. 4, a C 64-nél pedig 3 kilobyte). Nagyfelbontású grafikus üzemmódban az egyébként más célra felhasználható memória további 8 kilobytetal csökken. Megjegyzendő, hogy a C 64 társzervezése következtében a BASIC-interpreter számára csak mintegy 36 kilobyte (38911 byte) hozzáférhető, a többit csak gépi kódú rutinok támogatásával használhatjuk ki, míg a C 16-on – az említett 4 kilobyte kivételével – lényegében a teljes RAM-területet (64 kilobyte-os bővítő esetén mintegy 60 kilobyte-ot) használhatjuk BASIC-ből.
Mindkét gép társzervezése egyébként olyan, hogy közvetlenül csak 64 kilobyte címezhető. A memóriabővítések elve tehát mindig az, hogy ezen a címtartományon belül egyszer egyik, másszor másik memóriablokk aktiválódik. A PLUS/4 esetében ezek 16 kilobyte-os ROM és RAM blokkok, de vannak olyan megoldások is, ahol ezek csak 256 byte-os „ablakok". Az, hogy összességében hány kilobyte-nyi plusz memória keletkezik ily módon, már csak szervezés és szoftver kérdése.
A C 16-hoz kapható 64 kilobyteos tárbővítőkről minden számunkban olvashat hirdetéseket (a NEWLINE – 2220 Vecsés, Diófa u. 15. – 2000 Ft körüli áron ajánlja), az 1986/3 számunkban pedig tesztet is közöltünk az akkor kapható 5 típusról, sőt házilagos elkészítéséhez is adtunk útmutatást FUSI-rovatunkban.
Természetesen, a már említett technikai trükkök lehetővé teszik ennél nagyobb méretű memória kezelését is, így – elsősorban a C 64-hez – kaphatók 128, 256, 512 és 1024 Kbyte-os memóriabővítők is, meglehetősen borsos áron (15–40 ezer Ft körüli összegért). A PLUS/4-hez 128 kilobyte-os bővítőt 3160 Ft-ért a VIDEO ELEKTRONIKA GMK ajánl a C=újság 1987/5. számában. Ezek a RAM-ok már nem tekinthetők egyszerű memóriabővítésnek, mivel saját szoftverrel egészülnek ki, és elsősorban a viszonylag lassú lemezműveletek kiváltását célozzák. Néhány puffernyi memóriaterület feláldozása árán ugyanis a 1541-es vagy más Commodore lemezegységek működését szimulálják (RAM-diszk), de bennük a Ki/Be műveletek sokkal gyorsabban hajthatók végre, így az adatfeldolgozási folyamatok segítségükkel jelentősen meggyorsíthatók. Természetesen, végül az adatokat valódi lemezre kell másolni a RAM-diszkről. Használhatóságukat elsősorban a valamilyen lemezegységgel való kompatibilitásuk szabja meg.

4. Az esetleges különféle üzemmódokban mekkora számszerűen a képfelbontás, azaz hány sor és soronként hány képpont?

A képernyő-felbontás nagyfelbontású üzemmódban 320*200, többszínű üzemmódban 160*200 képpont, a karakteres képernyő pedig 40*25 karakterből áll. Ha a téma alaposabban érdekli, figyelmébe ajánljuk az 1987/3. és 1987/4. számunkban megjelent GRAFIKA A C 16-ON, PLUS/4-EN című cikkünket, illetve Vadnai Szabolcs: C 64 és C 16 Programozói Zsebkönyv című, megjelent idevágó műveit. A képfelbontás és képpont-leképezés kérdésében a három gép (C 64, PLUS/4, C 16) egyformán működik, míg a színinformáció leképezése és tárolása kérdésében már jelentős a különbség.

Két ügyben kérném eligazításukat:
1. Hogyan lehet BASIC programba TEDMON-t beiktatni?
A MONITOR parancsot még csak be lehet írni utasításnak, de ha ideér a program, akkor már nem ért BASIC-ül. Jó megoldást találtam lapjuk legutóbbi számában a Plus/4 karaktertervező programban: „print" utasítással írták be a „monitor" parancsot is, a rákövetkező TEDMON/parancsot és az „X"-t is. Működik is a program, bár nem értem, hogy hogyan. Megpróbáltam ezen a programon kívül is ugyanezt, de nem funkcionált. Miért?

A MONITOR egyes rutinjait SYS utasítással meg lehet hívni BASIC programból is, ha gondoskodunk a megfelelő paraméterek átadásáról. Erre jó példákat láthatunk a C= ÚJSÁG 87/6 számában megjelent SZUPER-PEEK programban. Sajnos ezzel a módszerrel nem használhatjuk ki a MONITOR összes lehetőségét, mert a rutinok egy része nem a hívó programnak adja vissza a vezérlést (nem RTS utasítással végződik), hanem a MONITOR parancsértelmező programjára ugrik.
A karaktertervező program a billentyűpufferbe írt CHR$(13) karakterekkel (ez a return billentyű kódja) hajtja végre a képernyőre printelt MONITOR parancsokat. A billentyűpuffer az 1319–1328 ($527–$530) címen található 10 byte. Ide POKE-ol az 1550-es sor egy HOME karaktert, az 1560-as sor FOR ciklusa pedig 9 db RETURN karaktert. A következő utasítás a 239-es címre beírja a puffer érvényes karaktereinek számát, ezután a program végrehajtása END utasítással befejeződik. Ekkor kezdődik meg a billentyűpuffer kiolvasása. A RETURN karakterek hatására az a képernyőre írt parancs kerül végrehajtásra, amelynek sorában a kurzor áll. A BASIC program szintén a képernyőre írt GOTO paranccsal folytatódik, ezért látszik úgy, mintha programból használtuk volna a MONITORT.

2. Ugyancsak a fent említett karaktertervező programban több ízben is előfordul ilyen utasítás: "IF P (2) THEN...". Ezt nem értem. Az IF-nek azt kellene megvizsgálnia, hogy valamilyen feltétel teljesül-e vagy sem. Egy csupasz változónév, minden reláció-megjelölés nélkül hogyan „feltétel"?
Tarnai Imre, 2645 Nagyoroszi, Szabadság tér 13.

Az IF utasítás valójában nem azt vizsgálja, hogy egy feltétel teljesült-e, csupán 0 vizsgálatot végez. Ha az IF után álló kifejezés értéke nem 0, akkor a THEN ágra kerül a vezérlés, ha 0 akkor az ELSE ágon (vagy a következő utasítással) folytatódik a program. Az interpreter kifejezéseket kiértékelő rutinja végzi el az összehasonlítást, ha egy kifejezésben reláció jelet talál (< > =). A feltétel teljesülése esetén a kifejezés értéke −1 lesz (mínusz 1), ellenkező esetben pedig 0. Ezt használjuk ki az IF utasításban a 0 vizsgálat rövidített írására. Tehát a fenti példa érthetőbben leírva: IF P(2) < > 0 THEN... A rövidítéssel nemcsak a program beírásánál takarékoskodunk, a végrehajtás is gyorsabb lesz, mert egyszerűbb kifejezést kell a programnak kiértékelni. A kifejezéseknek ezt a tulajdonságát arra is fel lehet használni, hogy egy változónak valamilyen feltételtől (feltételektől) függő értéket adjunk IF utasítás nélkül. Például az A=0−−(B=2)*13−(B=3)*3−(B>=4)**40 értékadó utasítás megfelel a következő programrészletnek:

```
10 A=0
20 IF B=2 THEN A=13
30 IF B=3 THEN A=3
40 IF B>=4 THEN A=40
```

Mindkét megoldás ugyanúgy működik. Hogy mikor melyiket használjuk, ez attól függ, hogy a program áttekinthetősége, vagy a végrehajtás sebessége a fontosabb.

Elnézést kérünk attól az olvasónktól, aki az iránt érdeklődött, hogy miként használhatná PLUS/4-es gépét saját építésű rádiótávcsövével gyűjtött észlelési adatainak frekvencia-elemzésére. Elképzelése szerint a távcsővel vett rádiósugárzást – megfelelő berendezések közbeiktatásával – a vételi frekvenciával arányos (0–9V) egyenfeszültséggé alakítaná, ezt mágnesszalagon rögzítené, majd megjelenítené a számítógép képernyőjén valamilyen formában. A levél szakértőtől szakértőhöz vándorolt, végül elveszett. De azért a válasz megszületett.

Nos, az elképzelés érdekes, és talán a C64-en – amelynek van analóg bemenete – megvalósítható is lenne a vázolt módon, de a PLUS/4-nél nincs analóg jelbeviteli lehetőség. Persze, ettől a feladat még nem válik megoldhatatlanná, csak másként kell a dolgot megfogni.

Javaslatunk a következő:

A vett rádiósugárzás frekvenciáját frekvenciaosztó berendezéssel kell olyan frekvenciamodulált négyszögjellé alakítani, hogy az eredő frekvenciatartomány 0 és kb. 2 Khz közé essen. Ha ugyanis figyelembe vesszük, hogy a normál szalagos bithossz 0.6–1.2 ms hosszú, továbbá, hogy a turbó-töltők ennél kb. 10-szer nagyobb jelsűrűséggel dolgoznak, adódik a becslés, hogy ilyen jelsűrűséget fel fogunk tudni dolgozni, de az is lehet, hogy még nagyobbat is. Mindenesetre a maximális feldolgozható frekvencia közvetlen összefüggésben van a mintavételi gyakorisággal. Az ily módon átalakított jel mágnesszalagra rögzítve és a szabványos magnóról (DATASETT) visszajátszva a normál magnóbemeneten keresztül beolvasható és feldolgozható lesz, a KERNAL interruptrutinjának és szalagbeolvasó rutinjainak alapos módosításával. Az alapelv az lehetne, hogy az időzítő-regisztereket a (tapasztalatilag meghatározandó, még megfelelően feldolgozható) legrövidebb értékre beállítva, mintavételezzük a bemeneti jelszintet. A fél-hullámhosszt a két jelszintváltás között eltelt idő adja. Minden egyes mintavételezéskor a RAM soronkövetkező bitje megkapná a jelszintnek megfelelő értéket (0/1), ami a feldolgozó rutin méretétől is függő, de egyszerre kb. 400 ezer észlelési adat memóriában történő átmeneti tárolását és feldolgozását tenné lehetővé, ami kb. 40–400 másodpercnyi szalagot jelent. A memória betelése után a szalag olvasását megszakítva, az adatokat fel lehet dolgozni, képernyőn megjeleníteni stb. Persze, olyan megoldás is elképzelhető, hogy a bejövő jel sűrűségeloszlását valamilyen módon közvetlenül a képernyőn rajzoljuk ki. Itt elsősorban sebességi korlátokkal kell számolnunk.

A szükséges programozási munka mikéntjére legjobb támogatást a ROM-lista, azon belül az interruptrutin ($ce00) és a szalag-olvasó rutinok ($e5fd, $e691 stb.) elemzése adhatja.

A képen látható „Mindenből a legjobbat" trikó a benne lévő hölgy nélkül – csak egyesületi tagok részére – ***megvásárolható!***

Megrendelésüket elküldhetik postán is. Ez esetben írják meg, hogy milyen méretet kérnek, s természetesen írják meg lakcímüket, tagsági számukat. Mi a megrendelés alapján utánvéttel küldjük el önöknek a kívánt trikót.

Az ára: 150 Ft/db.

Természetesen akik személyesen keresnek föl bennünket, azok a helyszínen maguk választhatják ki a megfelelő méretet.

Címünk postán is, személyesen is:

Commodore Egyesület
1136 Budapest Rajk László u. 15. II/3.
Telefonunk: 121-912
Minden nap 9–15 óra között várja önöket ezen a címen az Egyesület szervezője:
Winter Júlia



APOLLÓ – IBM

Legutóbb a számítógépek második generációjáról szóltunk, amelyeknek legfőbb elemei a tranzisztorok és a diódák voltak. E gépek korszaka elég gyorsan leáldozott, nemsokára követték őket a harmadik generációs gépek. Alig néhány évvel a tranzisztorok kifejlesztése után ugyanis kidolgozták az integrált áramköröket, amelyek a harmadik generációs számítógépek legfőbb építőelemeivé váltak. A harmadik generáció azután elég sokáig képviseltette magát, sőt e gépek közül sokkal még ma is találkozunk.

A számítógépeknek e korszakára jellemző az is, hogy mind jobban kettévált a gépek „csoda-számba menő", és mindennapos felhasználása. Az integrált áramkörök ugyanis már hatalmas sorozatban is gyárthatóak voltak, így azután a számítógépek mindinkább elterjedtek a hivatalokban, irodákban is.

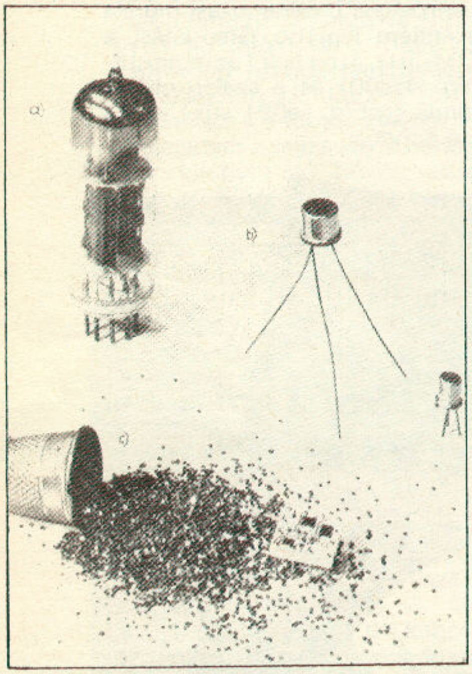
Arra pedig, hogy a legmagasabb szintű tudományos számításokban is használták e gépeket, egyik képünk mutat példát, amelyen az Apollo–13 irányítóterme látható, 1970-ből. A hatvanas évek végén, hetvenes évek elején ezek a számítógépek jelentették a csúcstechnológiát. Itt is olyan, IBM típusú gépeket használtak, amelyeknek alapegységei integrált áramköröket tartalmaztak.

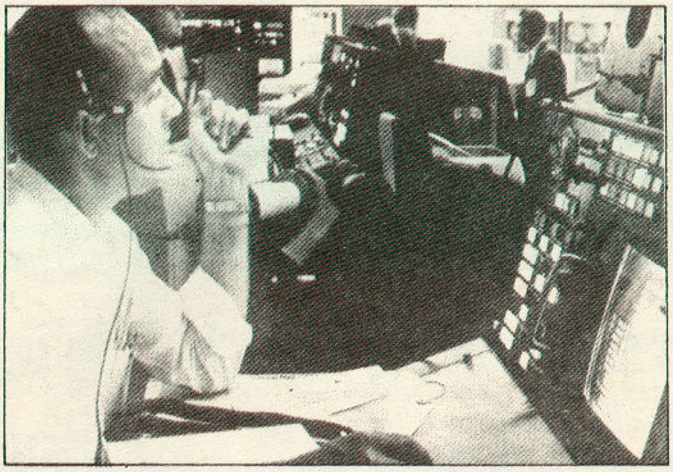
Hogy mi is az az integrált áramkör? Ez egy olyan áramkör, amelynek elemei ismét csak jóval kisebbek, mint az előző generációéi – azaz mint a tranzisztorok, vagy a diódák. Itt ugyan még szabad szemmel is megkülönböztethetők az alkotórészek, de már nem választhatók szét egymástól, mert önmagukban nem tudnának működni.

Az IC – vagyis az integrált áramkör – elemeit vagy egyetlen kristálylemezen alakítják ki különleges technikával, vagy pedig egy fényérzékeny, szigetelő lapkára először az összekötő elemeket nyomtatják rá – akár csak, mintha egy könyvet nyomtatnának –, majd a tranzisztorokat, diódákat hozzáforrasztják e laphoz. Nyilván ehhez arra is szükség van, hogy ezeknek az elemeknek a méreteit minimálisra csökkentsük. Hogy ez mennyire igaz, azt mutatja be képünk. Itt a gyűszűből kipotyogó szemcsék a harmadik generáció tranzisztorai – összehasonlításképpen pedig egy hagyományos tranzisztor és egy elektroncső látható.

A képen látható, az Apolló programban részt vevő gép olyan kiemelkedő lehetőségekkel rendelkezett, amelyeket ma is csak csodálhatunk. Ennek illusztrálására álljon itt egy-két adat: a teljes gép memóriakapacitása mintegy 200 Mbyte volt, és egyszerre negyven terminálon használhatták, mert már a multiprogramozásra is képes volt, azaz egy időben több program is futhatott rajta.

A multiprogramozás, illetve a több felhasználó általi egyidejű kezelhetőség egyébként a harmadik generációs számítógépek fontos sajátossága. Ezek miatt az előnyök miatt mind a mai napig megőrizték szerepüket a felhasználásban. Annak ellenére, hogy alkotóelemeik napjainkban már talán elavultnak tűnhetnek, időkihasználásuk és felhasználó-központúságuk még most is alkalmazhatóvá teszi őket.





Plus/4 térkép

Olvasónk nem mindennapi rajzzal állított be hozzánk. Egyetlen A/4-es lapra felvitte a Plus/4-es teljes kapcsolási rajzát. Minthogy e rajzra sok amatőr vágyik, úgy gondoltuk, nem sajnáljuk tőle a helyet. Sőt a jobb áttekinthetőség érdekében mi az A/3-as méret mellett döntöttünk. Először a szerző bevezetője:

A Commodore Plus/4 számítógép nemcsak a memória nagyságában és a beépített szoftvere miatt nyújt többet a C 16-os típushoz képest. Ugyanis van felhasználói kapuja (User Port) is, hasonlóan a C 64-hez.

A C 64-hez nagyon sok kiegészítő áramkör csatlakoztatható a felhasználói kapun keresztül. Ilyenek: EPROM-programozó, frekvencia mérő, A/D átalakító, adatgyűjtő kártya, feszültségmérő automatikus méréshatár váltással, logikai analizátor, stb.

A Plus/4-nél a $FD00–$FD0F tartományban találjuk meg a 6551 típusú soros kommunikációs interface adapter IC 4 regiszterét. A soros illesztés a felhasználói kapuján keresztül lehetővé teszi a nagy gépekhez vagy hálózathoz való csatlakozást V 24 vagy RS 232C szabvány szerint. (A 12 V-os jelszint helyett +5 V-os jelszinttel dolgozik.)

A soros illesztő időzítéseit a 6551-es IC állítja elő egy 1,8432 MHz-es kvarc időalappal. A soros illesztés eszközszáma 2-es. A felhasználói kapun van kivezetve és a $FD10–FD1F tartományban működtethető, egy 8 bites párhuzamos, kétirányú illesztést biztosító, 6529 B típusú IC. Ez speciális vezérlési feladatok megoldására ad lehetőséget.

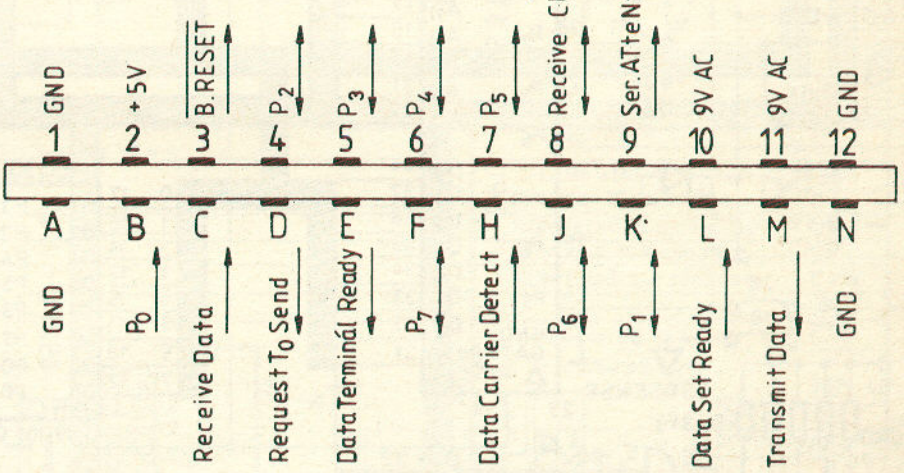
Ez a lehetőség ösztönzött a Plus/4 csatlakozási lehetőségeinek mielőbbi megismerésére. Sajnos a géphez nem adtak kapcsolási rajzot, így a legelső forrás a „C 16, Plus/4 programozói útmutató" melléklete volt. Ám örömöm hamar megcsappant, mert a mellékelt rajz hemzsegett a hibáktól. A csatlakozó rajzán pl. két M és két B volt. Ebből az egyik H, a másik 8 lehet. De melyik? És hol kezdődik a csatlakozó számozása? Amint utólag kiderült, nem vették át a C 64 csatlakozó jelöléseit, de még a bővítő azonos pontjait is másképp jelölték, mint a C 16-nál.

Ezért kénytelen voltam a rendelkezésemre álló rajzok (Plus/4, C 16) egyeztetése után saját szememmel meggyőződni a valóságról, majd az egészet egy lapra összerajzolni. A rajz áttekinthetősége miatt nem rajzoltam be a ki/bemeneteken lévő ferrit csövecskéket és az IC-k melletti tápfeszültség hidegítő kondenzátorokat.

Ezúton közreadom a „térképet", valamint a csatlakoztatáshoz nélkülözhetetlen rajzokat, amit később a javításhoz is felhasználható kiegészítésekkel teszek teljesebbé. **A csatlakozókat kívülről nézve rajzoltam!**

Zimányi Miklós

CN 2. USER PORT

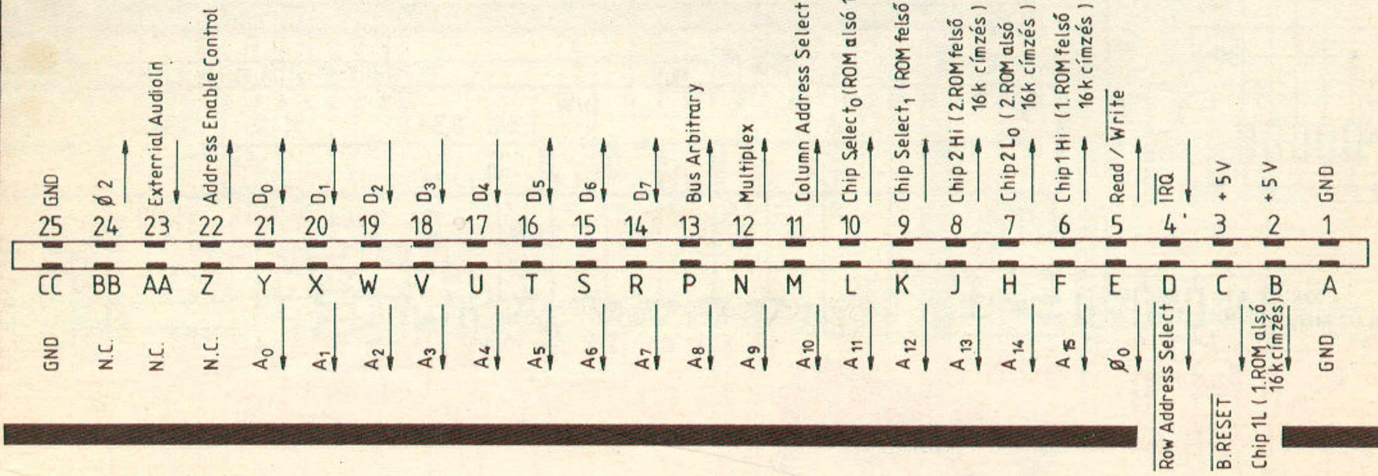


Csatlakozó:
TRW CHINCH CONN (3, 96 mm)
doubler., 0,2 row 2×12 pos
Molex Mod. 6511 pos. 12 Ord.No. 09-04-7121

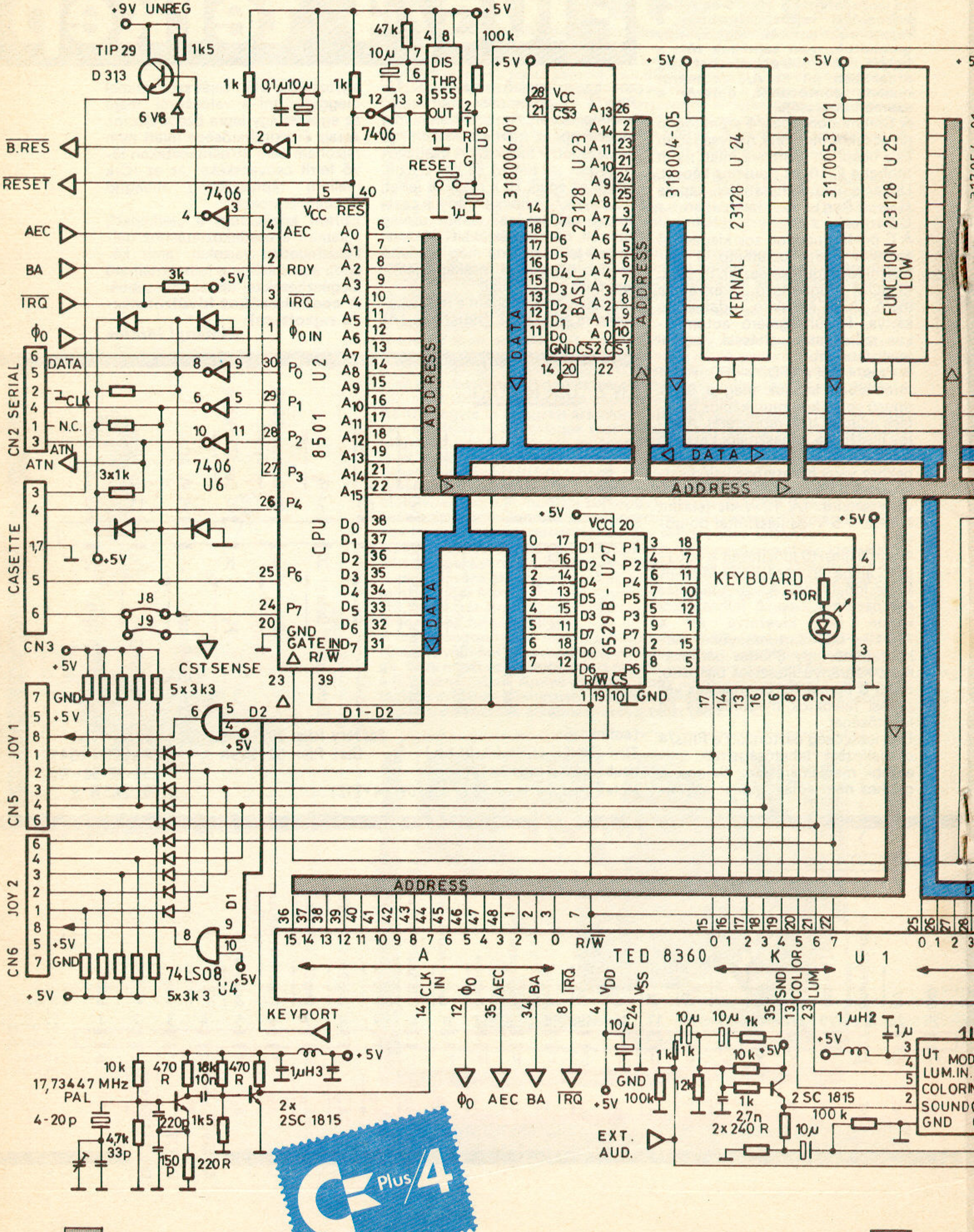
Factory P.No. 252-12-50-100
Dist. P.No. 50-245N-1

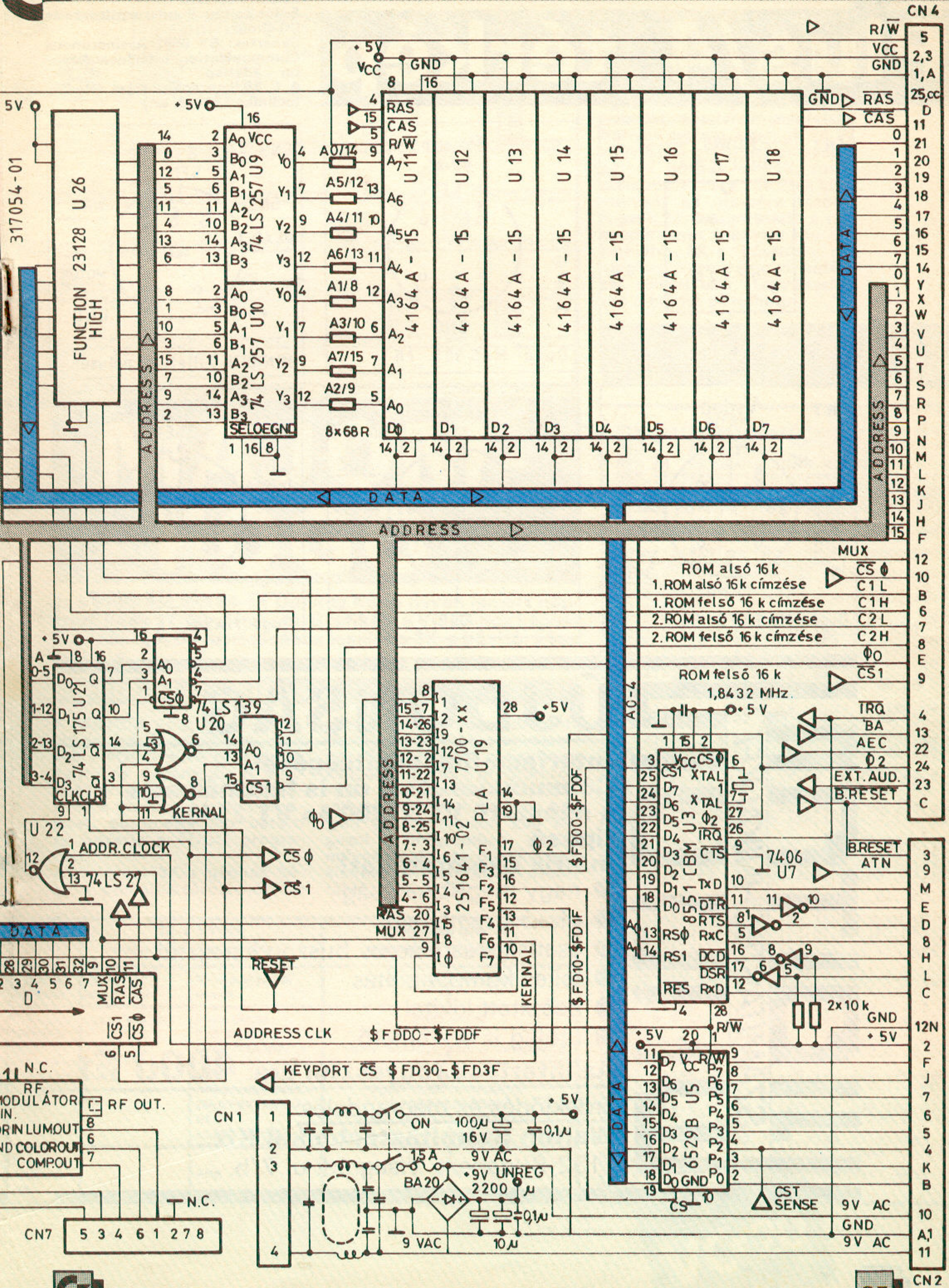
házzal: TRW-CARDCON 8545
251-12-50-170
50-245N-9

CN 4. MEMORY EXPANSION



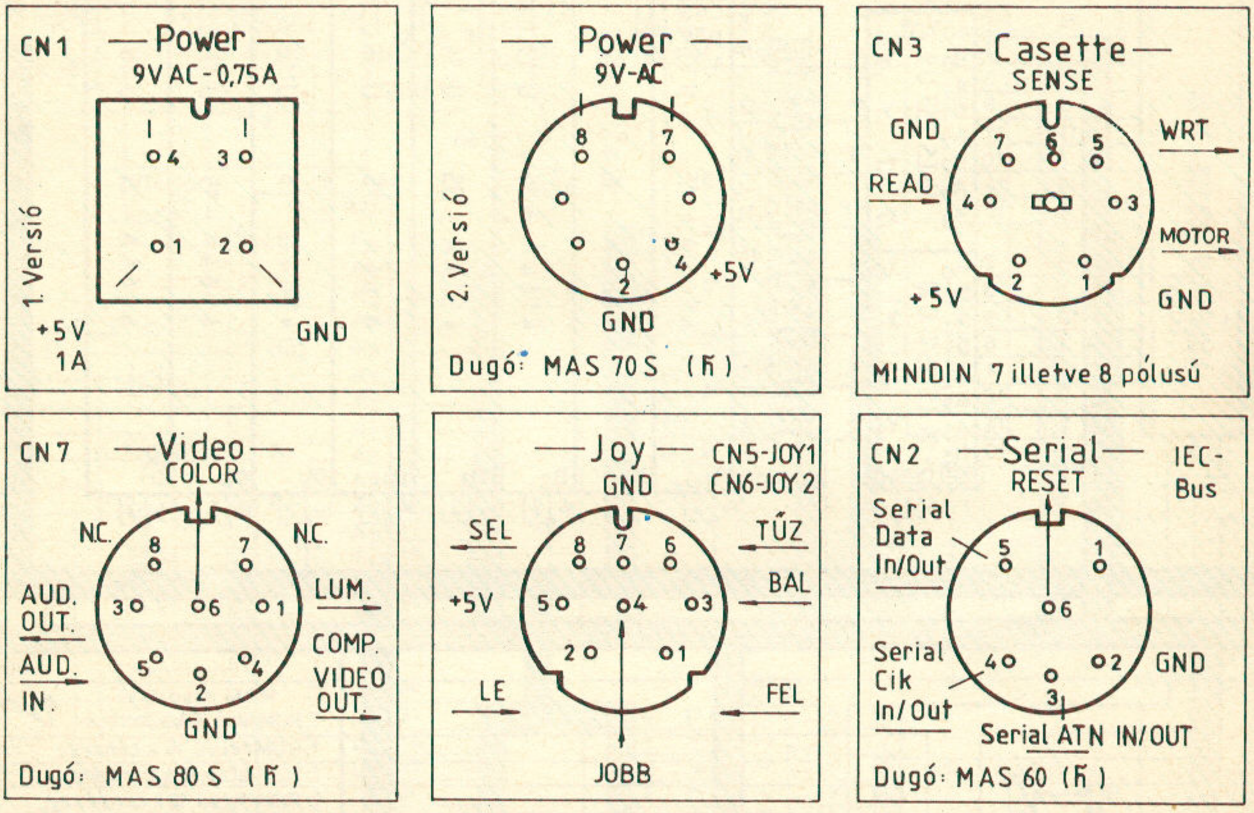
(illesztő kártya: 52 széles 1,4 vastag oszlop: ≈1,98 mm)

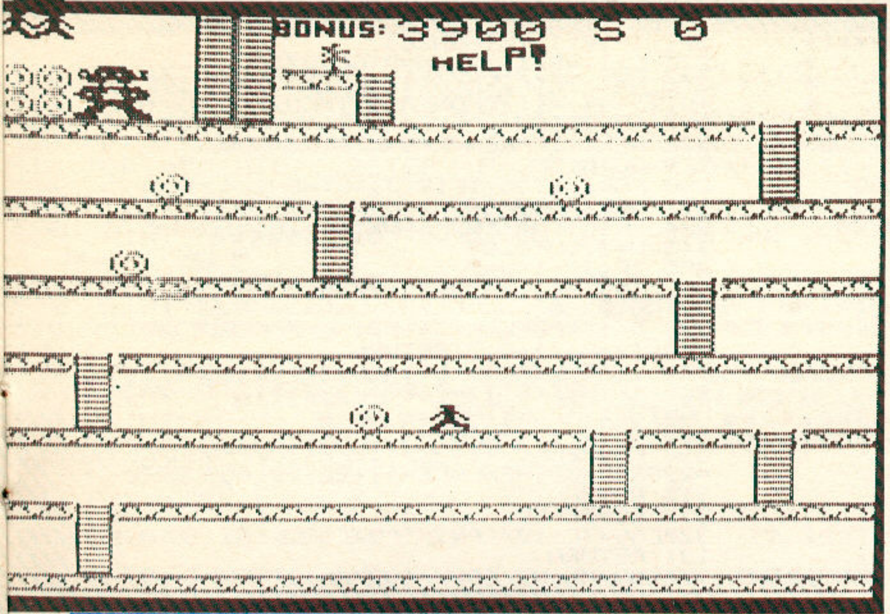

+9V UNREG
TIP 29
D 313
1k5
6 V8
B.RES
RESET
AEC
BA
IRQ
Φ0
CN2 SERIAL
DATA
CLK
N.C.
ATN
3x1k
CASETTE
CN3
J8
J9
CST SENSE
7406
U6
5x3k3
JOY 1
CN5
JOY 2
CN6
GND
+5V
74LS08
U4
D1
D2
KEYPORT
CPU
8501
U2
VCC
RES
RDY
Φ0 IN
GND
GATE IN
R/W
ADDRESS
DATA
D1-D2
47k
10µ
100k
555
DIS
THR
OUT
TRIG
U8
0,1µ
1µ
RESET
318006-01
BASIC
23128
U 23
VCC
CS3
GND
CS2
CS1
318004-05
KERNAL
U 24
3170053-01
FUNCTION LOW
U 25
6529B
U 27
KEYBOARD
510R
TED 8360
U 1
CLK IN
VDD
VSS
SND
COLOR
LUM
17,73447 MHz PAL
4-20p
10k
470 R
18k
10n
1µH3
4,7k
33p
220p
1k5
150 p
220R
2x 2SC 1815
100k
12k
2,7n
2x 240 R
2 SC 1815
EXT. AUD.
1 µH2
MOD
LUM.IN.
COLOR
SOUND
Plus/4


CN 4
R/W
VCC
GND
RAS
CAS
+5V
VCC
GND
317054-01
FUNCTION HIGH
23128 U 26
74 LS 257 U9
74 LS 257 U10
SELOEGND
8x68R
A0/14
A5/12
A4/11
A6/13
A1/8
A3/10
A7/15
A2/9
U 11
U 12
U 13
U 14
U 15
U 16
U 17
U 18
4164A - 15
D0
D1
D2
D3
D4
D5
D6
D7
DATA
ADDRESS
MUX
ROM alsó 16 k
1. ROM alsó 16 k címzése
1. ROM felső 16 k címzése
2. ROM alsó 16 k címzése
2. ROM felső 16 k címzése
ROM felső 16 k
1,8432 MHz.
CS ϕ
C1 L
C1 H
C2 L
C2 H
ϕ0
CS1
IRQ
BA
AEC
ϕ2
EXT. AUD.
B.RESET
ATN
74 LS 175 U21
74 LS 139 U 20
KERNAL
U 22
ADDR. CLOCK
74 LS 27
7700-xx
PLA U 19
251641-02
$FD00-$FD0F
$FD10-$FD1F
A0-4
8551 CBM U3
7406 U7
2x10 k
GND
+5V
RESET
ADDRESS CLK
$FDD0-$FDDF
KEYPORT CS $FD30-$FD3F
6529B U5
CST SENSE
N.C.
RF MODULÁTOR
RF OUT.
CN1
ON
100μ 16V
0,1μ
1,5 A
9V AC
BA20
+9V UNREG
2200 μ
0,1μ
10μ
9 VAC
9V AC
CN7
5 3 4 6 1 2 7 8
CN 2

Plus/4 térkép

A rajzok elkészítéséhez felhasználtam:
Tóth Viktor: A Commodore 16-os belső felépítése
Erdős Zoltán: Rendszerváltozók és I/0 címek
Synertek: SY 6551 Assincronous Communication Interface Adapter – adatlap
A C 16 kapcsolási rajza (Rádiótechnika 1986. 5. sz.)





A játékos feladata az óriási gorilla által fogva tartott lány megmentése. A figuránk az **A,W,S,Z** billentyűkkel irányítható. Az ugrásnál a **SHIFT**-tel együtt kell megnyomni azt a gombot, amelynek megfelelő irányba ugrani akarunk (A,W vagy S). A programban szerepelnek hordók, létrák, futószalagok, lift stb., de ezek kiismerését az olvasóra bízom. A szűk tárterület miatt a negyedik szint hiányzik, de aki legalább 8K-s bővítővel és megfelelő programozói tudással rendelkezik, természetesen a hiányt pótolhatja.

MAJOM-FOGÓ

Ez a program a C 64-en Donkey Kong néven ismert népszerű játék VC 20 változata. Grafikája a VC 20-hoz képest jó minőségű, de futtatásához legalább 3K-s bővítőre van szükség.

A PROGRAM BEÍRÁSA

A MAJOM program beírása, elmentése és kitörlése után kell a FOGÓ nevű programot beírni és elmenteni. A MAJOM program, miután előállította a grafikus karakterkészletet, automatikusan betölti és futtatja a FOGÓ programot.

A program 3K-s géphez készült, s magnós tároláshoz. De természetesen nagyobb bővítővel is futtatható, s ha valakinek floppyja van, annak sem kell lemondania a programról. Hogy ezekben az esetekben milyen teendőink vannak; milyen változtatásokat kell végezni a programban, illetve milyen direkt utasításokat kell kiadni a betöltéskor, azt a lenti táblázat mutatja:

	3K-s bővítő		nagyobb bővítő	
	magnóval	floppyval	magnóval	floppyval
Program beíráskor	minden a lista szerint	40-es sort ki kell hagyni	A 40-es sort ki kell hagyni	
1. program betöltése előtt	LOAD "MAJOM"	POKE 648,30 SYS 58648 LOAD "MAJOM",8	POKE 648,30 SYS 58648 LOAD "MAJOM"	LOAD "MAJOM",8
1. program betöltése után	RUN	RUN POKE44,33:POKE33•256,0:POKE56,64: NEW LOAD "FOGO",8	RUN POKE 44,33:POKE33•256,0: POKE 56,64:NEW LOAD "FOGO"	LOAD "FOGO",8
2. program betöltése után	–	RUN	RUN	

```
1 REM *********************************
2 REM *   C= UJSAG   SORSZAM: 075      *
3 REM *   MAJOM                        *
4 REM *   PROGRAM: TIHOR MIKLOS        *
5 REM *********************************
6 POKE36869,255:POKE37150,3:POKE788,194
7 PRINT"      DONKEY  KONG":PRINT"IR
TA TIHOR MIKLOS":PRINT"BALRA:A  JOBBR
A:S"
8 PRINT"FEL:W  LE:Z":PRINT"UGRAS: SHIF
T"
9 PRINT"START:RETURN":PRINT"KEREM VA
RJON":POKE36879,24:POKE650,128
10 FORI=7168TO7679:POKEI,PEEK(I+25600):N
EXT:POKE56,28:POKE52,28
20 FORI=0TO43:FORJ=0TO7:READS:POKE7168+8
*I+J,S:NEXTJ,I
30 FORI=7632TO7663STEP2:READA:POKEI,INT(
A/256):POKEI+1,A-256*INT(A/256):NEXT
35 FORI=7520TO7532STEP2:READA:POKEI,INT(
A/256):POKEI+1,A-256*INT(A/256):NEXT
40 PRINT"LOAD":POKE631,131:POKE198,1
100 DATA28,24,60,90,153,60,230,131,56,24
,60,90,153,60,103,193,28,24,60,60,60,24,
24,28
110 DATA56,24,60,60,60,24,24,56,152,88,6
3,24,24,62,99,192,25,26,252,24,24,124,19
8,3
120 DATA24,153,126,24,24,60,102,195,60,6
6,153,165,165,153,66,60,126,129,255,129,
255,129
130 DATA255,126,157,90,60,24,153,255,0,0
,185,90,60,24,153,255,0,0,129,255,129,25
5,129
140 DATA255,129,255,255,24,36,66,129,255
,0,0,24,112,62,88,60,126,16,24,7,111,93,
127
150 DATA30,4,63,126,224,246,186,254,112,
32,252,126,236,124,31,15,30,60,248,248
160 DATA55,62,248,240,120,60,31,31,1,3,3
1,23,31,31,63,127,192,224,240,92,244,152
,8,240
170 DATA255,255,255,255,127,61,61,125,22
4,252,254,198,240,248,216,224
180 DATA24,36,36,90,189,189,189,90,63,8,
5,2,2,5,8,63,252,16,160,64,64,160,16,252
190 DATA255,241,253,253,255,255,255,255,
0,255,195,36,60,102,102,126,16,124,254,1
86,16,16
200 DATA20,8,0,92,215,85,93,231,0,0,7,11
,61,75,255,128,127,63,0,192,232,124,255,
1,254
210 DATA252,0,63,12,3,3,12,63,0,0,0,0,0,
0,0,0,0,0,252,48,192,192,48,252,0,0,54,5
4,18
220 DATA36,0,0,0,255,255,170,255,85,85,8
5,255,16,48,56,108,102,54,26,12,8,12,28,
54,102
230 DATA108,88,48,0,238,170,234,170,238,
0,0,0,149,213,245,181,151,0,0,0,112,68,1
12,20,112
240 DATA0,0,0,0,0,7,148,247,148,151,0,1,
33,161,33,33,33,189,14,238,46,238,4,4,0,
4
300 DATA7705,7752,7711,7758,7717,7807,78
29,7895,7917,7961,7983,8027,8093,8115,81
59,7746
310 DATA7787,7842,7917,7968,8047,8051,81
00
```

VC=20

```
1 REM ***********************************
2 REM *     C= UJSAG SORSZAM 075         *
3 REM *     FOGO                         *
4 REM *     PROGRAM: TIHOR MIKLOS        *
5 REM ***********************************
10 O=30720:DIMA(8),B(8),IR(8),N(2),Z(2)
20 SC=0:MA=3
100 PA=0:GOSUB500:X=20:R=3:GOSUB3000
150 T=T+1:IFT=5THENT=0:BO=BO-100:PRINT"██"S
PC(9)BO"█ ":POKE7689,40:IFBO=0THEN700
152 GETA$:IFUG>0THEN235
153 IFA$=""THEN255
154 IFA$="W"ANDE=11THEN208
155 IFA$="Z"ANDPEEK(X+Y+22)=11THEN218
160 IFA$="S"ANDX<21ANDW=0THEN200
165 IFA$="A"ANDX>0ANDW=0THEN205
166 IFASC(A$)>192ANDASC(A$)<216ANDW=0ANDUG=
0THEN226
168 GOTO255
200 R=2:IFPEEK(X+Y)=2THENR=0
202 E1=1:E2=0:GOSUB250:GOTO255
205 R=3:IFPEEK(X+Y)=3THENR=1
206 E1=-1:E2=0:GOSUB250:GOTO255
208 IFR=5THENR=6:GOTO211
209 IFR=4THENR=5:GOTO211
210 R=4
211 E1=0:E2=-22:GOSUB250:IFY=7702ANDX<10THE
N2000
215 IFPA=1ANDY=7746THEN2100
216 IFE=11ANDPEEK(X+Y+22)=11THENW=1:GOTO255
217 W=0:GOTO255
218 IFR=6THENR=5:GOTO223
219 IFR=5THENR=4:GOTO223
220 R=6
223 E1=0:E2=22:GOSUB250:IFE=11ANDPEEK(X+Y+2
2)=11THENW=1:GOTO255
224 W=0:GOTO255
226 IFR=1ORR=3THENR=10:GOTO228
227 R=9
228 IFA$="o"THEN232
229 IFA$="♠"ANDX<21THENR=9:GOTO233
230 IFA$="♣"ANDX>0THENR=10:GOTO234
231 IFA$<>"♠"ANDA$<>"♣"THEN255
232 E1=0:E2=-22:UG=1:GOTO246
233 E1=1:E2=-22:UG=1:GOTO246
234 E1=-1:E2=-22:UG=1:GOTO246
235 GOSUB310
236 IFE1=0ANDUG=1THENUG=2:GOTO255
237 IFUG=2THENUG=0:E2=22:GOTO240
238 IFE2=-22THENE2=0:GOTO240
239 E2=22:UG=0
240 IFX=0ORX=21THENUG=0:E1=0:E2=22
242 IFUG>0THEN246
243 IFR=9THENR=2
244 IFR=10THENR=3
245 GOSUB250:GOTO255
246 GOSUB250:IFPEEK(X+Y+22)=12THENUG=0:E1=0
:E2=0:GOTO243
247 GOTO255
248 IFMM=1THENMM=0:GOTO700
249 MM=1:E1=0:E2=22:GOSUB250:GOTO255
250 POKEX+Y,E:POKEX+Y+O,6:X=X+E1:Y=Y+E2:E=P
EEK(X+Y)
251 IFE<>32ANDE<>11ANDE<>26ANDE<>27ANDE<>28
ANDE<>231THEN700
252 IFUG=0AND(PEEK(X+Y+22)=32ORPEEK(X+Y+22)
=231)THEN248
253 MM=0:IFE=26ORE=27THENE=28:GOSUB320
254 POKEX+Y,R:POKEX+Y+O,0:RETURN
255 ONPAGOTO350,400:L=L+1:IFINT(L/10)=L/10T
HEN280
256 IFM=1THENPRINT"██████NO███PQ":M=0
260 FORI=1TO8:IFIR(I)=0THEN278
262 POKEA(I)+B(I),P(I):IFP(I)=12THENPOKEA(I
)+B(I)+O,5:GOTO264
263 POKEA(I)+B(I)+O,6
264 ONIR(I)GOTO265,268,270
265 IFB(I)=8142ANDA(I)=21THENPOKEA(I)+B(I),
32:IR(I)=0:GOTO278
266 A(I)=A(I)+1:IFA(I)=22THENA(I)=21:B(I)=B
(I)+22:GOSUB300
267 GOTO276
268 A(I)=A(I)-1:IFA(I)=-1THENA(I)=0:B(I)=B(
I)+22:GOSUB300
269 GOTO276
270 GOSUB300:IFB(I)<YTHENIR(I)=3
271 IFIR(I)<3THEN264
272 B(I)=B(I)+22:N=8:GOTO277
276 N=7:IF(PEEK(A(I)+B(I)+22)=11ORPEEK(A(I)
+B(I)+44)=11)ANDX=A(I)THENIR(I)=3:N=8
277 GOSUB298
278 NEXT:IFPEEK(X+Y)<>RTHEN700
279 GOTO150
280 FORJ=1TO8:IFIR(J)=0THEN284
282 NEXT:GOTO260
284 PRINT"█████RS███TU":IFJ=1ORJ=5THENIR(J
)=3:N=8:GOTO286
285 IR(J)=1:N=7
286 I=J:A(I)=4:B(I)=7746:GOSUB298:M=1:P(J)=
32:GOTO260
298 P(I)=PEEK(A(I)+B(I)):POKEA(I)+B(I),N:PO
KEA(I)+B(I)+O,7:RETURN
300 IFPEEK(A(I)+B(I)+22)<>12THENRETURN
305 IFINT((B(I)-7680)/132)=(B(I)-7680)/132T
HENIR(I)=2:RETURN
308 IR(I)=1:RETURN
310 FF=PEEK(X+Y+22):IFFF=7ORFF=8ORFF=22ORFF
=29ORFF=30THENPOKEX+Y+22,28:GOSUB320
315 RETURN
319 SC=SC+BO
320 SC=SC+100:PRINT"██"SPC(16);SC:POKE7696,
131:RETURN
350 MN=MN+MI:IFMN=1ORMN=13THENMI=-MI
352 PRINT"████████████████████";MID$(SU$,MN,2
2)
355 IFPEEK(8039)=32THENPOKE8039,11:POKE3875
9,6
356 IFPEEK(8046)=32THENPOKE8046,11:POKE3876
6,6
358 IFY<>8032THEN362
360 IF(X=0ANDMI=-1)OR(X=21ANDMI=1)THEN700
361 E1=-MI:E2=0:GOSUB250
362 FORQ=1TO2:ONQGOTO363,368
363 IFZ(1)<7834THENZ(1)=Z(1)+1:GOTO374
364 POKEZ(1),32:Z(1)=Z(1)+1:IFZ(1)<7842THEN
373
366 POKEZ(1),32:Z(1)=7830:GOTO374
368 IFZ(2)>7854THENZ(2)=Z(2)-1:GOTO374
370 POKEZ(2)+1,32:Z(2)=Z(2)-1:IFZ(2)>7847TH
EN373
372 POKEZ(2)+1,32:Z(2)=7856:GOTO374
373 POKEZ(Q),29:POKEZ(Q)+O,7:POKEZ(Q)+1,30:
POKEZ(Q)+O+1,7
374 NEXT:POKE7835,11:POKE38555,6:POKE7854,1
1:POKE38574,6
375 IFY<>7834THEN380
376 E2=0:IFX<8THENE1=1:GOTO378
377 E1=-1
378 GOSUB250
380 IFPEEK(X+Y)<>RTHEN700
382 GOTO150
400 IFX<4ORX>5THENUP=0
401 IFX<11ORX>12THENDO=0
402 IFQ=2THENQ=1:GOTO404
403 Q=2
404 POKEZ(Q)+N(Q),231:POKEZ(Q)+N(Q)+O,6:POK
EZ(Q)+N(Q)+1,32:IFN(Q)=4THEN412
405 IFPEEK(Z(Q)+N(Q)-22)=RORPEEK(Z(Q)+N(Q)-
21)=RTHENDO=1
408 Z(Q)=Z(Q)+22:IFZ(Q)=8164THENZ(Q)=8142:N
(Q)=4
410 GOTO416
412 IFPEEK(Z(Q)+N(Q)-22)=RORPEEK(Z(Q)+N(Q)-
21)=RTHENUP=1
414 Z(Q)=Z(Q)-22:IFZ(Q)=7790THENZ(Q)=7812:N
(Q)=11
416 POKEZ(Q)+N(Q),12:POKEZ(Q)+N(Q)+1,12:POK
EZ(Q)+N(Q)+O,6:POKEZ(Q)+N(Q)+O+1,6
418 IFUP=1ANDZ(Q)=YANDN(Q)=4THENE1=0:E2=-22
:E=12:GOSUB250
420 IFDO=1ANDZ(Q)=Y+44THENE1=0:E2=22:GOSUB2
50
450 FORI=1TO4:S=256*PEEK(FA)+PEEK(FA+1):IFI
=1ORI=3THENGOSUB469:GOTO454
452 GOSUB468
454 POKE7706,11:POKE38426,6:POKES,32:POKES+
1,32:FA=FA+2:NEXT:IFFA=7664THENFA=7632
460 IFPEEK(X+Y)<>RTHEN700
461 GOTO150
468 POKES,31:POKES+1,33:GOTO470
469 POKES,23:POKES+1,24
470 POKES+O,7:POKES+O+1,7:RETURN
```

VC=20

```
500 PRINT"SJMMR":FORI=1TO6:PRINT"RLLLLLLLLLLLLLLLLLLLLLL":NEXT:GOSUB900
501 PRINT"SI      SKKQ&'(5000 SS"SC"Q";
502 IFSC<10000THENPRINT:PRINT"M";:IFSC>999THENPRINT"]";
505 PRINT"SQQQQKQQQQQ)*+":PRINT"SGGSNO SKKRLLSK":PRINT"SGGSPQ SKK  K"
510 FORI=7520TO7532STEP2:A=256*PEEK(I)+PEEK(I+1):FORJ=ATOA+44STEP22:POKEJ,11:POK
EJ+O,6:NEXTJ,I
515 FORI=1TO8:IR(I)=0:NEXT:RETURN
520 PRINT"SQQQQQQQQ&'(5000 SS"SC:PRINT"SM"SPC(6)"SRLLLLLLLLLL":PRINTSPC(5)"SS S
SNO    SS |"
521 PRINTSPC(5)"SS S      SPQ    SS |"
522 PRINT"RLSKRLLLLLLLLLLLLLLLLLLLSKRL";:FORI=1TO3
525 PRINT"QSK"SPC(18)"K":NEXT:PRINT"Q###SKQ##### S#VVX Q#####SKQ###RLLSKRLLLLL SS
SS ";
530 PRINT"SRLLLLLSKRLL";:FORI=1TO3:PRINT"QQSK"SPC(5)"S QQQ S"SPC(6)"K":NEXT
535 PRINT"TSCQQQQQNTZ":PRINT"RLLLRS QQ SSKRLLLLLSKRS QQ SLL";:FORI=1TO3
540 PRINTSPC(7)"SKQQQQQQK":NEXT:PRINT"Q#######################";
542 PRINT"RLLLLLLLLLLLLLLLLLLLLLL";:FORI=8057TO8145STEP22:FORJ=0TO15STEP5
543 POKEI+J,11:POKEI+J+O,6:NEXTJ,I
545 GOSUB900:POKE7954,26:POKE38674,2
546 FORI=7843TO7846:POKEI,22:POKEI+O,1:NEXT
548 SU$="S ]↑ ]↑   ]↑ ]↑         ]↑ ]↑   ]↑ ]↑ ":MN=1:MI=1:Z(1)=7930:Z(2)=7856:RET
URN
560 PRINT"SS     KK Q&'(5000 SS"SC:PRINT"SSS      SKKM"SPC(13)"SZ"
562 PRINT"  SNOSKKSLLSK"SPC(11)"SKSL  SPQSKK  K"SPC(11)"K SLLLLLLLLLLLLSKSL
 SK"
564 PRINT"     SLL     LL SK     K":PRINTSPC(14)"SK        SLL";:PRINTSPC(14)"SK  SL
L    SC";
566 PRINTSPC(11)"SLLSK":PRINTSPC(15)"SK":PRINT"SLLSK       KSLSK         K":FORI=1TO2:P
RINT"  SK     K K     K"
568 NEXT:PRINT"  K      K K     SL      SCSKSLL     SK K"SPC(7)"SLL":PRINT"SK       K
 K"SPC(10)"SLSK";
570 PRINT"SK          K K"SPC(11)"KK         KSZSK"SPC(11)"KSLLSK     SLLL"SPC(11)"SK  K
";
572 PRINTSPC(17)"SLL  SK"SPC(14)"SLL       SK"SPC(11)"SLL"SPC(6)"SLLL LL       LL";
574 FORI=7816TO8146STEP22:POKEI,231:POKEI+7,231:POKEI+O,6:POKEI+O+7,6:NEXT
576 Z(1)=8010:Z(2)=7988:N(1)=4:N(2)=11:FA=7632
590 RETURN
700 N=E:FORI=YTO8142STEP22:POKEI+X,N:POKEI+X+O,2:N=PEEK(I+X+22):POKEI+X+22,R
710 POKEI+X+O+22,0:NEXT:POKE38883+X,2:POKE38885+X,2
720 MA=MA-1:POKE7679+MA,32:IFMA=0THENPRINT"SQQQQQQQQQQQQQQQQQQQVEGE":GOTO1000
730 FORI=1TO500:NEXT:ONPA+1GOTO100,2010,2105
900 FORI=8164TO8185:POKEI,12:POKEI+O,5:NEXT:RETURN
1000 IFPEEK(197)<>15THEN1000
1005 RUN
2000 PRINT"SM"SPC(8)"";:IFPA=0THENPRINT"M";
2001 PRINT"QCSS#    Q":GOSUB319:FORI=1TO2000:NEXT:IFPA=2THEN100
2005 PRINT"SM"SPC(9);"QCSS#    Q":FORI=1TO2000:NEXT
2010 PA=1:GOSUB520:X=20:R=3:GOSUB3000:GOTO150
2100 GOSUB319:FORI=1TO2000:NEXT
2105 PA=2:GOSUB560:X=0:R=2:GOSUB3000:GOTO150
3000 Y=8142:E=32:E1=0:E2=0:W=0:UG=0:L=-1:POKEX+Y,R:POKEX+Y+O,0:BO=5000:IFMA=1THE
NRETURN
3002 FORI=7680TO7678+MA:POKEI,0:POKEI+O,0:NEXT:RETURN
```

FIGYELEM!

A MAJOM program lefuttatása után a gép karakterkészlete megváltozik. A további POKE-ok és LOAD-ok beírásakor, tehát szinte vakon kell dolgoznunk! Hiszen az egyes betűk helyett mindenféle krikszkrakszok jelennek meg a képernyőn. S minthogy a listában az átlagosnál sokkal többféle grafikus karakter olvasható (vagy éppen nem olvasható), közöljük ezeknek a karaktereknek a helyét a billentyűzeten.

Jó játékot kíván a szerző:

Tihor Miklós

CTRL+1	"[illegible]"
CTRL+2	"[illegible]"
CTRL+3	"[illegible]"
CTRL+4	"[illegible]"
CTRL+5	"[illegible]"
CTRL+6	"[illegible]"
CTRL+7	"[illegible]"
CTRL+8	"[illegible]"
CTRL+9	"[illegible]"
CTRL+0	"[illegible]"
LE	"[illegible]"
FEL	"[illegible]"
JOBBRA	"[illegible]"
BALRA	"[illegible]"
HOME	"[illegible]"
CLEAR	"[illegible]"

commo-do-re

Ezt a programot egy időre „jegeltük". Pár hónapja már nálunk van, de a szerzők (a lapjainkon már bemutatott Kővári és fia) beleegyezésével előbb egy kicsit gondolkodtunk, nem kellene-e inkább forgalmazásra eladni ezt a zeneszerkesztőt. Őszintén szólva ma is azt gondoljuk, hogy találtunk volna rá forgalmazót, de végül is győzött a szerkesztői ambíció, hiszen ilyen kvalitású programot nem mindennap közölnek a hazai szaklapok. Aki egy kicsit is érdeklődik a zeneprogramozás iránt, annak feltétlenül meg kell rendelni, vagy be kell pötyögnie a Commo-do-re-t!

A PROGRAM BEGÉPELÉSE

Aki vállalkozik arra, hogy bepötyögje a programot, annak a következőket javasoljuk.

Először gépeljük be a gépi nyelvű betöltőt. A begépelés után „RUN"-nal indítsuk el. Ha hiba nélkül lefut, akkor mentsük el. Ha adathibát jelez, akkor javítsuk ki a hibát, és újból kíséreljük meg a futtatást.

Ha elmentettük a betöltőt, akkor gépeljük be a „BASIC" részt. Hibátlan begépelés után a programvég mutatók helyes értéke:

PEEK(46) = 45
PEEK(45) = 27

A BASIC programrész begépelésénél ne kísérletezzünk a futtatással, amíg nincs meg a gépi nyelvű rész is!

Most parancs módban írjuk be a következő sort.

CLR: FOR I = 2048 TO 11547: CS = CS + PEEK(I): NEXT: PRINT CS

Az **ellenőrző összeg** helyes értéke **701876.**

Ha rendben van, akkor mentsük el a „BASIC" részt is. A két részletet a következő módon kapcsolhatjuk össze.

Először betöltjük a gépi betöltőt, majd „RUN"-nal elindítjuk. Megvárjuk az átkódolást, majd ha a „READY" megjött, akkor betöltjük a „BASIC" részt. Ezután beírjuk a programvég mutatókat.

POKE 46,50
POKE 45,235

Az így összekapcsolt programot futtatás nélkül mentsük el. Ha ezután visszatöltjük, akkor már együtt lesz a teljes program. Aki a „PÖTYÖGŐ SZOLGÁLATNÁL" megrendeli a programot, annak semmi más tennivalója sincs, mint „LOAD"-olni és „RUN"-nolni.

FIGYELEM!

Lapunk történetében ez a második program, amely két változatban rendelhető meg, ezért van a programnak kettős sorszáma:

– A COMMO-DO-RE 071 azonosítójú program gépi nyelvre fordított, a futása gyorsabb. (Igaz a gyorsabb futás, csak a zenelistázásnál vehető észre).

– A COMMO-DO-RE 070 azonosítójú program szerkesztő részének nagy része viszont „BASIC" nyelvű.



EGY KIS ELMÉLET

A következő kis kitérővel nem elriasztani akarjuk az elméletet nem kedvelőket a COMMO-DO-RE használatától. Épp ellenkezőleg. Bizonyos, a programban használt fogalmakat próbálunk megmagyarázni, valamint olyan segédértékeket adni, amelyeket az is felhasználhat, aki az elméletre egyébként nem kíváncsi.

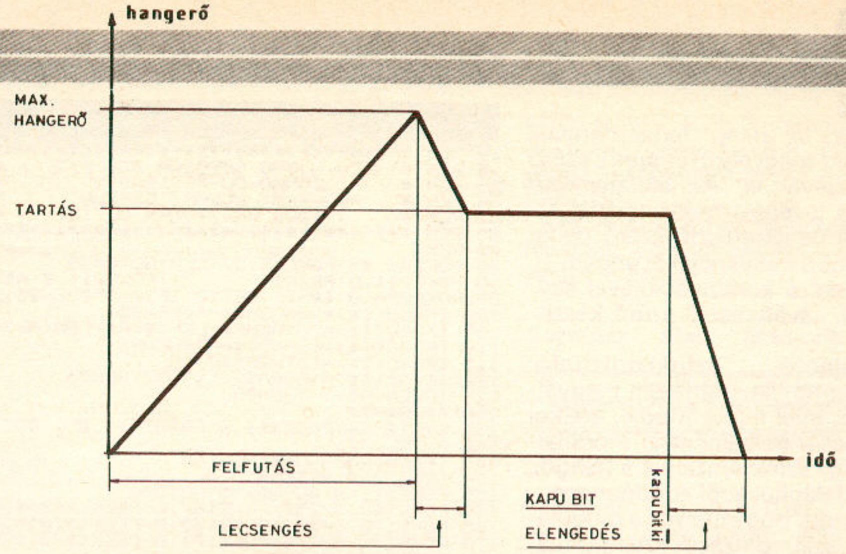
A hang meglehetősen összetett jelenség. Érzékelni tudjuk magasságát, hangszínét, hangerejét és időtartamát.

A hang magassága a frekvenciájával jellemezhető. A zenei gyakorlatban azonban nem így adjuk meg a hang magasságát, hanem oktávval és hanggal.

A hang színezetét, hangzását a különböző felharmonikus frekvenciák aránya határozza meg. A felharmonikus tartalom szoros összefüggésben van a megszólaló hang idő függvényében ábrázolt alakjával. A C-64 hangelőállító chipje, a SID négyféle hullámalakot képes generálni. Ezek a következők: **háromszög, fűrész, négyszög és zaj.** Az egyes hullámalakok – a zaj kivételével – kombinálhatók egymással, tehát egyszerre több hullámalak is kiválasztható. A háromszög jel kevés felharmonikust tartalmazó, színtelen hangot ad. Pl. fuvola. A fűrészjel felharmonikusokban gazdag, zengő hangot ad. Pl. csemballó. A négyszög jel felharmonikus tartalma a kitöltési tényezőtől függ. A kitöltési tényező a 0 és 1 szintek időtartamának arányát jelenti. A zaj hullámalakkal pl. az ütőhangszereket utánozhatjuk. A hullámalakok a SID kontroll regisztereiben (rövidítve C.R.) egy-egy bit beállításával aktivizálhatók. A kontroll regiszterek legkisebb helyiértékű bitje a kapubit. Ennek szerepére a későbbiekben még visszatérünk.

A hang erőssége a megszólalástól az elhallgatásig nem állandó. A hangerő időbeni változását a mellékelt ábrán figyelhetjük meg. Az Y tengelyen két hangerő szintet különböztethetünk meg. A maximális hangerőt – amit csak egyszerűen hangerőnek nevezünk – és az ún. tartási szintet. Az X tengelyen négy időintervallumot különíthetünk el. Az első szakaszban a hangerő 0-tól a maximális értékig fut fel. Ennek a szakasznak a neve **FELFUTÁS.** A következő sza-

kaszban a maximálisról a tartás szintjére esik a hangerő. Ez az időtartam a **LECSENGÉS**. Mindaddig a **TARTÁS** szintjén marad a hangerő, amíg a kontrollregiszterben a kapubitet 0-ba nem állítjuk. Ezután az **ELENGEDÉS** ideje alatt a tartás szintjéről 0-ra csökken a hangerő. Ezeket az adatokat az eredeti angol szavak rövidítéseiből **ADSR értékeknek** nevezzük. Mindegyik érték 0 és 15 közötti szám lehet. Az ADSR értékek jellemzőek egy-egy hangszerre. Segítségül közöljük néhány hangszer jellemző ADSR értékeit és hullámalakját:



1. TÁBLÁZAT

Hangszer	Hullámalak	Felfutás	Lecsengés	Tartás	Elengedés
Zongora	Négyszög	0	9	0	0
Furulya	Háromszög	6	0	0	0
Csembaló	Fűrészfog	0	9	0	0
Xylofon	Háromszög	0	9	0	0
Orgona	Háromszög	0	0	15	0
Akkordeon	Háromszög	6	6	0	0
Trombita	Fűrészfog	6	0	0	0

```
100 REM*********************************
110 REM*  C= UJSAG SORSZAM:070-071     *
130 REM*  COMMO-DO-RE GEPI BETOLTO     *
150 REM*  PROGRAM: KOVARI LASZLO       *
160 REM*********************************
165 REM ***  ELLENORZO OSSZEGEK  ***
170 DATA 2219,2257,2668,2351,2502,2178
180 DATA 2258,2385,2569,1446,2046,1927
190 DATA 1697,2050,2136,1940,1897,2487
200 DATA 1676,1882,2326,2060,1879,2669
210 DATA 2348,2062,1930,1813,1299,2235
220 DATA 2027,1821,1765,2013,1407,2136
230 DATA 2169,1770,2317,2624,2116,2159
240 DATA 2011,1874,1785,2279,1905,2475
250 DATA 1646,1830,1945,2403,2244,2210
260 DATA 2034,2010,2403,1645,2343,2420
270 DATA 2255,2235,2156,2155,2071,1575
280 DATA 2578
285 REM ***         ADATOK         ***
290 DATA FKMKFMIKAJJCANIAFKLKFMHKAPACAJ
300 DATA PBAKAABLLKBJHKGOHKANCAGOIKGOLK
310 DATA ANCAGOMKFKLKFMJKANKOFKMKFMKKAN
320 DATA EOAGFKLKANCAGMMKGMLKFKJKANCAGM
330 DATA KKGMJKFKJKIDFOLKFINKFKKKFOMKFI
340 DATA OKFKNKIBFGHKFIHKFKOKFGIKFIIKFK
350 DATA HKANCAGMIKGMHKAKAABLJKBJHKFKHK
360 DATA ANCAGMIKGMHKFKJKANCAGMKKGMJKFK
370 DATA JKFMLKANGOFKKKFMMKANAOAGKIIEIJ
380 DATA IEAKAANKMADAADCAIMIMJKAAFINIFI
390 DATA OIGONIANCAGOOIIGKKIGJNLIAAANJA
400 DATA IEKIJNMIAAANEAAPBDIEKIIEGOLIAN
410 DATA CAGOMICKAABKLIJMAJANNAJKDAIBFG
420 DATA LIFILIAJCAGOMIANLMJMAIANEAJKCA
430 DATA ANNOJMPPANJAJKAAFILIFIMIIGIGAG
440 DATA NKMADAJCPHANJNJKBAFIMMGEPMEKDN
450 DATA BLBNJCPHOKHICAMEDBKOAKAABLFKJM
460 DATA AIANADIMBLFKIDFOHKKKIMBLFKFOIK
470 DATA IKAJEBFKLKABCDFKJKAKBAIBBHFKBJ
480 DATA FKJKAAIMBHFKBJFKJKDAIBFGFKFIFK
490 DATA AJCAGOGKANIMJMAJANEAJKEAANNOJM
500 DATA PPANBAAGGKLKIOKIJCHAANAOIJAKBA
510 DATA FMKKAPEBALJBJKDAGKLKIBJGDAKMAN
520 DATA KPBJFKKIIMBJFKANDMKIFMJKAJJOBL
530 DATA FKIDFOJKBJFKIMBLFKFOKKBJFKIBAJ
540 DATA NKACENADJKJAFIDNFKFMJMEAAPDEJM
550 DATA BAANFAJKAAFIALAGFKLKJMBAANHOFK
560 DATA FKIBJGKOKKFKGKJGCDIKNKNADAFILI
570 DATA NKOADAFIMIFKHKJAAINIMADAACEIPC
580 DATA GKNIFKOIACNMNLJKACACCNPPACCNPP
590 DATA ACCNPPIBAJDLFKLKJMEAAJAMJKEAFI
600 DATA LKANKLIHJKAGFIFKACFKAAKLNLAABA
610 DATA FIGKKMNLAABAFIFKJKLKIBFGFKAJCA
620 DATA GOGKFIFKNKEBDAFIHPNKFBDAFIIPFK
630 DATA GKNIFBDAFKFKNIEBDAIBJGNEAJCAGO
640 DATA GKFIELGOGKFIFKFKGKFIFLJKIBIBFG
650 DATA FKAJCAGOGKFIFKAKAABLFKFIHKIMCK
660 DATA AAKIIEAPIAJKAAIBJGHAKMANKPKKBL
670 DATA FKNJFAENIMBLFKNJGAENIMBLFKFJKK
680 DATA IGKKIMIOEOHKANLNBLFKNIIBENIMBL
690 DATA FKFIJKIMIJIBFGFKAJCAGOGKIDFOEL
700 DATA FIFKFKGKFOFLFIGKFKFKIDJOIBALCA
710 DATA GMGKFIFKJKBAFIOKJKCAFILKFIALIF
720 DATA AGAKAAFKALAPHEJMBAANGBFKHKJMBA
730 DATA APKAJMCAAPDAMICBENMILAENMIEAEN
740 DATA MGHPAAGMLKANJPFKELIBFGFKFIFKFK
750 DATA FLFGGKFIGKJKIBIBFGFKAJCAGOGKFI
760 DATA FKBLFKJMPPAPLAJMAIAPJCJMAJAPCE
770 DATA IBAJIFAKAAFKHKJMBAAPKAJMCAAPDA
780 DATA MICBENMILAENMIEAENFKHPNIEBDAFK
790 DATA IPNIFBDAMGHPAAFKOKANGAIMIMIMIB
800 DATA AJDMGMOKIMBLFKIEIMBLFKFIGKIGFI
810 DATA FKAKAAIBAJIJIMBLFKFIJPIMBLFKFI
820 DATA KPIMIJIEBLFKAKAABJJPIGIKIMIBAJ
830 DATA HJCKAABLFKJMAKAPPFIJIEKIIEAPJA
840 DATA JKAAIBJGHAKMANKPKKIEFLKKJCOPNJ
850 DATA EAENBLFKANGAFIJPFIKPAPJCJMALAN
860 DATA GAIGIBAJCDKOKOKEKEKEKEKKBLFKJC
870 DATA PAKAIKBLELFIJPIMBLELFIKPAOHAAP
880 DATA HAGEKPGGJPIOANFPIGKKFKJPNJAAEN
890 DATA FKKPNJBAENFLKKNJEAENIGKKIGIKIM
900 DATA IOEOHKANFJBLFKANGAJKBAFILKANNA
910 DATA GKJKKMAPGAIBBHFKIBAJHPFILKIMIJ
920 DATA IBFGFKAJCAGOGKIDFOELFIFKFKGKFO
930 DATA FLFIGKFKFKIDJOIBALCAGMGKFIFKMG
940 DATA HPAAAAAAOBGIIBOILIGJOHPJKPIKGA
950 DATA DLMKNLDPIMGOENPIBOIPOOOCNPAAAA
960 PRINT"□"TAB(13)"KEREM VARJON !"
970 DIMEL(67):FORI=1TO67:READEL(I)
980 CS=CS+EL(I):NEXTI
990 IFCS<>140003THEN1070
1000 E=1:FORI=12032TO13034STEP15:READA$
1010 CS=0:FORD=1TO30STEP2
1020 A=ASC(MID$(A$,D,1))
1030 A=A-65+(ASC(MID$(A$,D+1,1))-65)*16
1040 CS=CS+A:POKEI+D/2,A:NEXTD
1050 IFCS<>EL(E)THEN1090
1060 E=E+1:NEXTI:END
1070 A$=" ELLENORZO OSSZEG HIBA!"
1080 PRINTA$:END
1090 A$=" HIBAS ADAT A"
1100 PRINTA$280+E*10"■. SORBAN !"
1110 GOTO1060
```

Szűrés: A hang felharmonikus tartalmát megváltoztathatjuk szűrő segítségével is. Az aluláteresztő szűrő a levágási frekvenciánál kisebb, a felüláteresztő szűrő pedig a nagyobb frekvenciákat engedi át. A két szűrő kombinációjával sávzáró ill. sáváteresztő szűrő készíthető.

Moduláció, szinkronizáció: A felharmonikus tartalom megváltoztatásának másik módja, amikor egy másik frekvenciával moduláljuk, vagy szinkronizáljuk a hangot. A SID felépítéséből adódóan meghatározott, hogy melyek a moduláció ill. szinkronizáció szempontjából összetartozó csatornák:

2. TÁBLÁZAT

Csatorna	Szinkronizáló, ill. moduláló csatorna
1	3
2	1
3	2

A PROGRAM HASZNÁLATA

A program indítás után felajánlja a menüt. Az inverz alakban megjelenő kezdőbetűkkel lehet választani.

SZERKESZTÉS

Ha új művet szeretnénk „komponálni", tehát nincs a memóriában zeneadat, akkor legelőször meg kell adni a szólamok számát, majd ki kell tölteni a megjelenő regisztertáblázatot. Alaphelyzetben minden regiszterben 0 van, a tempó értéke 1. A kurzorvezérlő billentyűkkel lehet a regisztereket elérni. A regiszterek gyakorlatilag azonosak az előzőekben – az elméleti részben – ismertetett fogalmakkal. Tehát a táblázat sorrendjében:

HANGERŐ – TEMPÓ

Majd szólamonként:

FELFUTÁS – LECSENGÉS – TARTÁS – ELENGEDÉS – HULLÁMALAK

Figyelem!

A regisztertáblázatot nagyon gondosan kell kitölteni. Lehet olyan kombináció is, ahol meg sem szól a csatorna.

A regisztertáblázat kitöltése után megjelenik az 1-es sorszám, mellette egy V betű látható. Ez gyakorlatilag a tárban lévő zeneprogram

```
10 REM *********************************
15 REM *    C= UJSAG SORSZAM 070-071    *
20 REM *       COMMO-DO-RE  BASIC      *
30 REM *    PROGRAM: KOVARI LASZLO     *
32 REM *       1202 BP. TOMPA U. 18.   *
35 REM *********************************
40 REM:
80 POKE46,45:POKE45,27:CLR:GOTO445
90 POKE211,B:PRINT"      ";:POKE211,B:A$=""
95 POKE204,0:I$="":GETI$:IFI$=""THEN95
100 IFASC(I$)=20ANDA$=""THEN95
105 IFASC(I$)=20THENA$=LEFT$(A$,LEN(A$)-1):GOTO135
110 IFASC(I$)=13ANDA$=""THEN95
115 IFASC(I$)=13THEN140
120 IFASC(I$)<48ORASC(I$)>57THEN95
125 IFLEN(A$)>4THEN95
130 A$=A$+I$
135 SYS12264:POKE211,B:PRINTA$" ▌";:GOTO95
140 A=VAL(A$)
145 IFA<MINORA>MAXTHEN155
150 SYS12264:RETURN
155 POKE211,B:GOSUB205:GOTO90
160 POKE168,D/256:POKE167,D-PEEK(168)*256
165 POKE170,B/256:POKE169,B-PEEK(170)*256
170 POKE172,A/256:POKE171,A-PEEK(172)*256
175 SYS12032:RETURN
180 B=CI+3*(SZAND7)+2:POKE140,B/256:POKE139,B-PEEK(140)*256:B=15
185 POKE780,SZ:POKE782,A/256:POKE781,A-PEEK(782)*256:SYS12164
190 A=PEEK(139)+256*PEEK(140)+1:RETURN
195 C=C+1
200 POKE211,18:A=PEEK(C):PRINTA:RETURN
201 C=C+.5
202 POKE211,18:A=PEEK(C):IFC=INT(C)THENPRINTINT(A/16):RETURN
203 PRINTAAND15:RETURN
205 SYS12264:POKE211,B:PRINT"            ";
210 POKE54296,15:POKE54273,100:POKE54278,240:POKE54276,17
215 FORK=0TO50:NEXT:POKE54296,0:RETURN
220 PRINT"■";:GOSUB225:PRINT"≡";:RETURN
225 SYS12264:A$=A$+I$:POKE211,B:PRINTA$"    ■■▌";:POKE204,0:RETURN
230 POKE211,B:POKE204,0:A$=""
235 I$="":GETI$:IFI$=""THEN235
240 IFA$=""THEN300
245 IFASC(I$)=20THEN295
250 A=ASC(I$)
255 IF(AAND127)<65OR(AAND127)>72ORA=66ORA=200ORA=197ORA=194THENGOSUB210:GOTO235
260 IF(AAND128)=128THENI$=CHR$(AAND127)+"#"
265 GOSUB225
270 A=VAL(A$)
275 GETI$:IFI$=""THEN275
280 IFASC(I$)=20THEN295
285 IFASC(I$)=13THENA$=RIGHT$(A$,LEN(A$)-1):SYS12264:RETURN
290 GOSUB210:GOTO275
295 A$=LEFT$(A$,(LEN(A$)>1)*-1):I$="":GOTO330
300 IFI$="Z"THENGOSUB220:A=0:GOTO275
305 IFI$="O"THENGOSUB220:A=160:GOTO275
310 IFI$="K"THENGOSUB220:A=176:GOTO275
315 IFI$="U"ANDC=GTHENA=128:GOSUB220:GOTO275
320 IFI$="M"ANDC=GTHENA=144:GOSUB220:GOTO275
325 IFASC(I$)<48ORASC(I$)>55THENGOSUB210:GOTO235
330 GOSUB225:GOTO235
335 D=I:PRINT"≡"I;:FORJ=1TOSZ:A=PEEK(C)
340 POKE211,J*6+1:IFA>0ANDA<128THEN355
345 ON(A=255)*-1+(A=176)*-2+(A=128)*-3+(A=144)*-4GOTO365,370,375,395
350 ON(A=0)*-1+(A=160)*-2GOTO405,410
355 PRINT"≡"(AAND240)/16"▌"H$(AAND15);
360 C=C+1:NEXT:POKE211,J*6+1:PRINT"≡"PEEK(C):RETURN
365 J=4:NEXT:I=I+20:PRINT"■ V":RETURN
370 PRINT"■ K";:GOTO360
375 PRINT"■ U     ";
380 SZ=SZOR128:A=PEEK(C+1)+256*PEEK(C+2)+CI-1:C=C+2:GOSUB180
385 SZ=SZAND127:PRINTPEEK(141)+256*PEEK(142)
390 J=4:NEXT:RETURN
395 PRINT"■ M      "PEEK(C+1)+256*PEEK(C+2);:POKE211,19:PRINTPEEK(C+3)
400 C=C+3:GOTO390
405 PRINT"■ Z";:GOTO360
410 PRINT"■ O";:GOTO360
415 POKE198,1:POKE631,ASC(I$):RETURN
420 POKE13010,X/256:POKE13009,X-PEEK(13010)*256:RETURN
425 PRINT"◙":INPUT"NEV";B$:B$="●"+B$:PRINT"▌EGYSEG SZAM ?";
435 MIN=1:MAX=15:B=14:GOSUB90:PRINT"▌"
440 RETURN
442 PRINT"ADATATVITELI HIBA !"
443 GETA$:IFA$=""THEN443
444 CLOSE1:GOTO460
445 POKE53280,12:POKE53281,12:DIMH$(11):FORJ=0TO11:READH$(J):NEXT
455 DATA C;C#,D,D#,E,F,F#,G,G#,A,A#,H
460 CI=13035:B=19:C=CI:X=PEEK(13009)+PEEK(13010)*256:SZ=PEEK(CI)
465 PRINT"◙■■KOVARI L.      COMMO-DO-RE  .   BP. 1987.▌▌▌≡"
470 PRINT"■S■ZERKESZTES▌"
475 PRINT"■M■EGHALLGATAS▌"
480 PRINT"■K■IMENTES▌"
485 PRINT"■B■ETOLTES▌"
490 PRINT"■O■SSZEKAPCSOLAS▌"
495 PRINT"■H■ E L P"
500 GETA$:IFA$=""THEN500
505 IFA$="S"ANDX=0THEN550
510 IFA$="B"THEN1190
515 IFA$="H"THEN1320
520 IFX=0THEN500
525 IFA$="S"THEN560
530 IFA$="M"THEN1305
535 IFA$="K"THEN1175
540 IFA$="O"THEN1210
545 GOTO500
550 PRINT"◙ SZOLAM (1-3)";:MIN=1:MAX=3:GOSUB90
555 POKEC,A:FORI=C+1TOI+A*3:POKEI,0:NEXT:POKEI,1:POKEI+1,255:X=I+1:GOSUB420
560 H=0:PRINT"◙ SZOLAM";:GOSUB200:SZ=A
562 C=CI+SZ*3+1:D=0:PRINT"▌ HANGERO (0-15)";:GOSUB200
564 PRINT" TEMPO    (0-15)";:GOSUB195:C=C-1
566 PRINT"▌▌":J=0
567 POKE211,19:POKE204,0
568 GETI$:IFI$=""THEN568
569 FORI=0TO50:NEXT
```

```
570 IFI$="]"THEN581
572 IFI$="▓"THEN620
574 IFI$="T"THEN1000
576 IFI$="L"THEN680
577 IFI$="V"THENSYS12264:GOTO460
578 IFI$="]"THENSYS12264:X=0:GOSUB420:GOTO460
579 IF(I$<"0"ORI$>"9"ORJ<>0)AND(I$<>"N"ANDI$<>"I"ORJ=0)THEN568
580 GOTO640
581 IFD=0THEN590
582 IFC=CI+(D-1)*3+1THEND=D-1:GOTO594
584 IFJ=0THEN587
585 J=J/2+(J=16)*8:IFJTHEN588
586 SYS12264:PRINT"TTTT"
587 C=C-.5
588 SYS12264:PRINT"]";:GOTO567
590 IFC=CI+SZ*3+1THEN568
592 C=C-1:GOTO588
594 IFD=0THENC=CI:GOTO560
595 IFD=SZ+1THENC=C+3:GOTO705
596 SYS12264:PRINT"]"TAB(14);D".SZOLAM▓"
598 C=CI+(D-1)*3+1:PRINT" FELFUTAS   (0-15)";:GOSUB202
600 PRINT" LECSENGES (0-15)";:GOSUB201
602 PRINT" TARTAS     (0-15)";:GOSUB201
604 PRINT" ELENGEDES (0-15)";:GOSUB201
606 C=C+.5:PRINTTAB(55)"HULLAMALAK▓"
608 PRINT" HAROMSZOG:"
610 PRINT" FURESZFOG:"
612 PRINT" NEGYSZOG:"
614 PRINT" ZAJ:TTTT"
616 FORK=4TO7:PRINTTAB(19)MID$("NI",(PEEK(C)AND2↑K)/2↑K+1,1):NEXT
618 C=CI+(D-1)*3+1:GOTO566
620 SYS12264:IFD=0THEN630
622 IFJ=128THEN632
624 SYS12264:IFJ>0THENJ=J*2:GOTO628
626 C=C+.5:IFC=CI+D*3THENJ=16:PRINT"▓▓"
628 PRINT:GOTO567
630 IFC=CI+SZ*3+1THENC=C+1:GOTO628
632 D=D+1:GOTO594
640 IFI$>="0"ANDI$<="9"THEN648
642 SYS12264:PRINTI$;:IFI$="I"THENPOKEC,PEEK(C)ORJ+1:GOTO644
643 POKEC,(PEEK(C)AND255-J)OR1
644 IFJ=128THEN567
646 GOTO624
648 MIN=0-(C=CI+SZ*3+2):MAX=15:GOSUB415:GOSUB90:IFD=0THEN654
650 IFC=INT(C)THENPOKEC,(PEEK(C)AND15)+A*16:GOTO626
652 POKEC,(PEEK(C)AND240)+A:GOTO626
654 POKEC,A:IFMIN=0THENC=C+1:GOTO628
655 GOTO567
680 SYS12264:PRINT"▓":FORI=0TO3:PRINTTAB(200):NEXT:PRINT"▓LISTA:"
685 PRINT"SORSZAM ?";:MIN=0:MAX=65535:GOSUB90
690 C=CI:IFA=0THEN560
695 I=A:GOSUB180:IFA=1THENGOSUB155:GOTO690
700 C=A:GOTO710
705 I=1
710 PRINT"]":H=I
715 FORI=ITOI+19:G=C
720 GOSUB335:C=C+1:NEXT:C=G:I=D:F=D
725 E=-(PEEK(G)=128)-(PEEK(G)=144)*2
730 PRINT"▓]";:POKE211,((E=2ANDC=G+3)+C-G+1)*6+2
735 POKE204,0:GETI$:IFI$=""THEN735
740 J=0:IFI$>="0"ANDI$<="9"THEN775
745 IF(I$="Z"ORI$="O"ORI$="K"OR(I$="U"ORI$="M")ANDC=G)AND(E=0ORC=G)THEN780
750 ON-(I$="▓")-(I$="]")*2-(I$="▓")*3-(I$="▓")*4GOTO895,950,885,940
755 ON-(I$=CHR$(13))-(I$="↑")*2-(I$="B")*3GOTO970,975,1055
760 IFI$="L"ORI$="V"THEN576
765 IFI$="T"THEN1000
770 GOSUB210:GOTO735
775 IFEANDC>GTHENGOSUB415:ONE-(C-G>1)GOTO840,865,880
780 B=(C-G+1)*6+2:GOSUB415:IFC=G+SZANDE=0THENGOTO830
785 GOSUB230:IFPEEK(G)<255THEN985
790 J=J+1
795 IFA$=""THEN810
800 FORD=0TO11:IFA$<>H$(D)THENNEXT
805 A=A*16+D:D=12:NEXT
810 POKEC,A:IFJ=0THEN885
815 B=B+6:ON-(A=128)-(A=144)*2GOTO835,860
820 C=C+1:IFJ<SZTHENGOSUB230:GOTO790
825 POKEC+1,255:X=C+1:GOSUB420
830 B=SZ*6+8:MIN=0:MAX=16:GOSUB90:POKEC,A:GOTO885
835 C=C+1:POKEC+2,255:X=C+2:GOSUB420
840 B=14:MIN=1:MAX=65535:GOSUB90
845 GOSUB180:B=14:IFA=1THENGOSUB155:GOTO845
850 A=A-CI:POKEC+1,A/256:POKEC,A-PEEK(C+1)*256
855 GOTO895
860 C=C+1:POKEC+3,255:X=C+3:GOSUB420
865 B=14:MIN=0:MAX=65535:GOSUB90
870 POKEC+1,A/256:POKEC,A-PEEK(C+1)*256:IFJ=0THEN885
875 C=C+2
880 MIN=0:MAX=255:B=20:GOSUB90:POKEC,A:GOTO895
885 IFEANDC>=G+EORE=0ANDC=G+SZORPEEK(G)=255THEN895
890 C=C+1-(E>0ANDC=G+1):SYS12264:PRINT:GOTO725
895 IFPEEK(G)=255THEN770
900 I=I+1:GOSUB935
905 IFPEEK(214)<20THENPRINT:GOTO915
910 POKE781,1:SYS59903:SYS59626:F=F+1:H=H+1
915 K=0
920 PRINT"]":GOSUB335:C=G:I=D
925 IFKTHENC=K+(PEEK(G)=128)
930 GOTO725
935 A=I:GOSUB180:C=A:G=C:SYS12264:RETURN
940 IFC>GTHENC=C-1+(E=2ANDC=G+3):SYS12264:PRINT:GOTO725
945 K=C-1:GOTO955
950 K=0
955 IFI=1THEN770
960 I=I-1:GOSUB935:IFI>=HTHEN925
965 POKE781,20:SYS59903:POKE781,1:SYS59752:POKE218,132:H=H-1:F=F-1:GOTO920
970 SYS12264:A=F:GOTO695
975 SYS12264:A=H-19:IFA<1THENA=1
980 GOTO695
985 D=-(A=128)-(A=144)*2
990 IFD=ETHEN795
995 GOSUB210:PRINT"]":GOSUB335:I=D:C=G:GOTO730
1000 SYS12264:PRINT"▓":FORJ=0TO3:PRINTTAB(200):NEXT
```

végét jelenti. Nyilván, amint az 1-es sorba zeneadatokat írunk, a végjel automatikusan a 2-es sorba tolódik.

Elkezdhetjük tehát a zeneadatok bevitelét. Egy sor beírása 3 szólam esetén a következő:

Sorszám	Első szólam	Második szólam	Harmadik szólam	Ütem
1	3C	4C#	5C	4
2	0	5G	Z	4

A program az oktávot 0 és 7 közötti számként, a hangokat pedig a zenei hangok betűjelével kezeli. A használható hangok: **C,C#,D,D#,E,F,F#,G,G#,A, A#,H.**

A félhangokat a programban SHIFT-tel kell megadni, de a képernyőn SHIFT + C C#-ként jelenik meg. A hangokat mindig oktávval együtt kell beírni. Pl. 5C jelentése: 5. oktáv C hangja.

A sor elé a sorszám automatikusan kiíródik. A kurzorvezérlő billentyűkkel bármelyik adat elérhető és javítható. A „RETURN" billentyűvel lehet lapozni a listában előre, a „↑" billentyűvel pedig vissza.

A hangok helyett további utasítások is megadhatók.

Z – szünet: hatására a kiválasztott hang nem szól.

O – változatlan hang: hatására a hang úgy szól tovább, ahogy eddig.

K – kapu bit ki: hatására a tartási szintről az elengedés sebességével csökken a hangerő.

U – ugrás: csak a sor első helyén fogadja el a program. Az „U"-t egy létező sorszámnak kell követnie. Hatására a program megvizsgálja az ugrást engedélyező regiszter tartalmát, és ha ez nem 0, akkor a zene a megadott sorszámnál folytatódik és az ugrást engedélyező regiszter tartalma 1-gyel csökken. Az ugrást engedélyező regiszter címe 174, a zene hívásakor tartalma 1, tehát egy ugrás engedélyezett. Hogy mindez miért jó így? Tegyük fel, hogy zenénk első tíz hangját egyszer akarjuk megismételni. Nyilván a tizenegyedik hang helyére beírjuk az ugrás utasítást. Amikor a zene lejátsszásakor a program ideér, visszamegy az első hangra, s ismét lejátsza az első tíz hangot. Tizenegyedikként újra megtalálja az ugrás utasítást, s ha most is 1-et találna az ugrást engedélyező regiszterben, megint viszszamenne az első hangra. S ez így menne a végtelenségig. Természetesen „kompozíciónk" további részeiben ismét szükség lehet az ugrásra. Ehhez az M utasítás segítsé-

gével kell átállítani a regiszter tartalmát 1-re, vagy akár többre is! Ha azonban nagyobb értéket írunk a regiszterbe, vigyázzunk, nehogy valahol végtelen ismétlésbe keveredjünk!

M – memóriába írás: csak a sor első hangja helyén használható. Az „M"-et két számnak kell követnie. Az első szám a memória egy címét jelenti, s így értéke 0 és 65535 között lehet. A második szám a beírandó adatot jelenti, ezért ez 0 és 255 között lehet. Az „M" utasításokat szabadon elhelyezhetjük a zene adatai között, ezek a zenében nem hallhatók. Ezekkel a műveletekkel írhatjuk be a SID regisztereibe a szükséges adatokat (kitöltési tényező, szűrőadatok stb.), vagy kapcsolatot létesíthetünk segítségükkel a zeneprogram és a főprogram között.

Figyelem!

Vannak olyan adatok, amelyeket csak memóriába írással lehet (vagy kell) betölteni. Pl. kitöltési tényező, szűrőadatok stb.

L – listázás: hatására a képernyő alján megjelenik a „LISTA:" felirat és a „SORSZÁM" kérdés. Most meg kell adni a kívánt sorszámot, amelytől a listát kérjük. Az új lista legalsó során helyezkedik el a kurzor. Ha 0-át adunk, a regisztertáblázatot listázza.

T – törlés: törlés kérésekor meg kell adni az első és utolsó törlendő sor számát. Ha első sorszámnak 0-át adunk, akkor visszaadja az eredeti listát. Ezután a megadott sorokat a program törli, majd automatikusan átsorszámozza a listát. A programban szereplő ugrás műveleteket úgy korrigálja, hogy az eredeti helyekre mutassanak. Ha olyan sort töröltünk, amelyre valamelyik ugrás művelet vonatkozott, akkor ez az ugrás az 1-es sorszámra fog mutatni.

B – beszúrás: bármelyik sorszám elé beszúrhatunk egy új sort. A kurzort arra a sorra kell vinni, ahová be akarunk szúrni, majd „B". A képernyő alsó részén megszerkesztjük az új sort. A program beszúrja az új sort, átsorszámozza a listát és korrigálja az ugrás műveleteket.

SHIFT+CLR – az egész zene törlése: csak akkor fogadja el a program, ha a regisztertáblázat van a képernyőn.

V – visszatérés a menübe.

Figyelem!

Szerkesztés üzemben mindig be kell fejezni az elkezdett sort. Ha hibát vétettünk, azt csak az aktuális hangnál tudjuk javítani. A sor befejezése után viszont bármelyik sor javítható.

```
1005 A=65535:GOSUB180:MIN=0:MAX=256*PEEK(142)+PEEK(141)-1
1010 PRINT"■TORLES: ELSO SORSZAM ?";:B=23:GOSUB90
1015 IFA=0THENA=H:GOTO690
1020 MIN=A:GOSUB180:B=23:D=A:PRINT
1025 PRINTTAB(46)"UTOLSO SORSZAM ?";
1030 GOSUB90:A=A+1:GOSUB180
1035 B=X+1:GOSUB160:X=X-(A-D):GOSUB420
1040 B=CI+3*(SZAND7)+3:POKE166,B/256:POKE165,B-PEEK(166)*256:POKE171,SZ
1045 B=D-CI:POKE168,B/256:POKE167,B-PEEK(168)*256:B=A-D:POKE170,B/256
1050 POKE169,B-PEEK(170)*256:SZ=SZAND7:SYS12282:A=H:GOTO690
1055 IFG=XTHEN770
1060 SYS12264:PRINT"■":FORJ=0TO3:PRINTTAB(200):NEXT
1065 PRINT"■BESZURAS:":PRINT:PRINTI;
1070 C=G:FORJ=1TOSZ
1075 B=(C-G+1)*6+2:GOSUB230:IFA$=""THEN1120
1080 FORK=0TO11:IFA$=H$(K)THENA=A*16+K
1085 NEXT
1090 POKE251+J-1,A:C=C+1:NEXT
1095 B=(C-G+1)*6+2:MIN=0:MAX=16:GOSUB90
1100 POKE251+J-1,A
1105 A=G:B=X+1:D=G+J:GOSUB160
1110 FORK=0TOJ-1:POKEG+K,PEEK(251+K):NEXT
1115 X=X+J:GOSUB420:J=A:A=D:D=J:SZ=SZOR128:GOTO1040
1120 IFA=0ORA=160ORA=176THEN1090
1125 IFJ>1THEN1075
1130 J=SZ+1:NEXT:J=0
1135 POKE251,A:IFA=144THEN1155
1140 A=65535:GOSUB180:MIN=1:MAX=256*PEEK(141)+PEEK(142)
1145 GOSUB90:GOSUB180:A=A-CI:POKE253,A/256:POKE252,A-PEEK(253)*256
1150 J=3:GOTO1105
1155 MIN=1:MAX=65535:B=14:GOSUB90
1160 POKE253,A/256:POKE252,A-PEEK(253)*256
1165 MIN=0:MAX=255:B=20:GOSUB90
1170 POKE254,A:J=4:GOTO1105
1175 GOSUB425:OPEN1,A,1,B$
1180 POKE252,CI/256:POKE251,CI-PEEK(252)*256:POKE780,251:A=X+1
1185 POKE782,A/256:POKE781,A-PEEK(782)*256:SYS65496:IFST=0THENCLOSE1:GOTO460
1186 GOTO442
1190 GOSUB425:OPEN1,A,0,B$:POKE780,0:POKE782,CI/256:POKE781,CI-PEEK(782)*256
1200 SYS65493:X=256*PEEK(782)+PEEK(781)-1:GOSUB420
1205 IFST=64THENCLOSE1:GOTO460
1206 GOTO442
1210 PRINT"■"TAB(44)"KEREM A PROGRAM VEGE MUTATOKAT!"TAB(91)"PEEK(45)?";
1215 MIN=0:MAX=255:B=24:GOSUB90
1220 C=A:PRINTTAB(91)"PEEK(46)?";:MIN=8:GOSUB90
1225 D=C+256*A:B=X+1:A=CI-535:C=D+B-A
1230 PRINTTAB(87)"A ZENE HOSSZA:"B-A"BYTE"
1235 PRINTTAB(87)"A ZENE HIVASA:SYS"D
1240 PRINTTAB(91)"OSSZEKAPCSOLAS:"
1245 PRINTTAB(87)"POKE46,"INT(C/256)":";
1250 PRINT"POKE 45,"C-INT(C/256)*256
1255 PRINTTAB(96)"■RETURN■"
1260 FORI=0TO131:POKE828+I,PEEK(12032+I):NEXT
1265 FORI=960TOI+5:READJ:POKEI,J:NEXT
1270 DATA 32,60,3,76,226,252
1275 POKE168,D/256:POKE167,D-PEEK(168)*256
1280 POKE170,B/256:POKE169,B-PEEK(170)*256
1285 POKE172,A/256:POKE171,A-PEEK(172)*256
1290 GETA$:IFA$=""THEN1290
1295 IFASC(A$)<>13THEN1290
1300 SYS960
1305 FORJ=54272TOJ+24:POKEJ,0:NEXT
1310 B=CI+3*SZ+2:POKE782,B/256:POKE781,B-PEEK(782)*256
1315 PRINT"■■SORSZAM:";:SYS12408:PRINT"■":POKE198,0:GOTO460
1320 PRINT"■■ M ■ + CIM + ADAT   *   MEMORIABA IRAS■"
1330 PRINT"■ U ■ + SORSZAM       *   UGRAS■"
1335 PRINT"■ V ■"TAB(23)"VISSZA A MENUBE■"
1340 PRINT"■ L ■"TAB(23)"LISTA■"
1345 PRINT"■ T ■"TAB(23)"TORLES■"
1350 PRINT"■ B ■"TAB(23)"BESZURAS■"
1355 PRINT"■ Z ■"TAB(23)"SZUNET■"
1360 PRINT"■ O ■"TAB(23)"VALTOZATLAN HANG■"
1365 PRINT"■ K ■"TAB(23)"KAPU BIT KI■"
1370 PRINT"■CLR■  (REG.TABL.-NAL)   EGESZ ZENE TORLES"
1375 PRINT"■RETURN■"TAB(23)"LAPOZAS ELORE■"
1380 PRINT"■ ↑ ■"TAB(23)"LAPOZAS HATRA■"
1385 PRINT"* CSAK ELSO HANGNAL ADHATO!";
1390 GETA$:IFA$=""THEN1390
1395 PRINT"■"TAB(14)"CIMEK:■"
1400 FORK=1TO3
1405 PRINTK"■.SZOLAM FREKV.L:"TAB(28)54265+K*7
1410 PRINTK"■.SZOLAM FREKV.H:"TAB(28)54266+K*7
1415 PRINTK"■.SZOLAM IMP.SZEL.L:"TAB(28)54267+K*7
1420 PRINTK"■.SZOLAM IMP.SZEL.H (4 BIT)"54268+K*7
1425 PRINTK"■.SZOLAM KONTROLL REG.:     "163+K*7
1430 PRINTK"■.SZOLAM F/L:"TAB(28)54270+K*7
1435 PRINTK"■.SZOLAM T/E:"TAB(28)54271+K*7;:IFK<3THENPRINT"■"
1440 NEXT
1445 GETA$:IFA$=""THEN1445
1450 PRINT"■"TAB(12)"CIMEK:■"
1455 PRINT"SZUROFREKV. L: (3 BIT)   . 54293"
1460 PRINT"SZUROFREKV. H:"TAB(26)54294
1465 PRINT"R3;R2;R1;R0 | F0;F3;F2;F1  54295"
1470 PRINT"3KI;F.SZ;S.SZ;A.SZ | HE.   54296■"
1475 PRINT"TEMPO (1-15):"TAB(26)169
1480 PRINT"UGRAS ENGEDELYEZO REG.:      174"
1485 PRINT"STOP REGISZTER :             176"
1490 GETA$:IFA$=""THEN1490
1495 PRINT"■"TAB(8)"HASZNALHATO HANGOK:■"
1500 PRINT"C     ■ C ■          C#    ■SHIFT C ■■"
1505 PRINT"D     ■ D ■          D#    ■SHIFT D ■■"
1510 PRINT"E     ■ E ■■"
1515 PRINT"F     ■ F ■          F#    ■SHIFT F ■■"
1520 PRINT"G     ■ G ■          G#    ■SHIFT G ■■"
1525 PRINT"A     ■ A ■          A#    ■SHIFT A ■■"
1530 PRINT"H     ■ H ■■"
1535 PRINT"  MINDEN HANGOT OKTAVVAL KELL MEGADNI!■"
1540 PRINTTAB(10)"FORMATUM 3 SZOLAMRA:■"
1545 PRINT"1.SZ.    2.SZ.    3.SZ.   UTEM■"
1550 PRINT" 3C       4C       5C       8■"
1555 GETA$:IFA$=""THEN1555
1560 GOTO465
```

MEGHALLGATÁS

Ezt a menüpontot csak akkor fogadja el a program, ha van zeneadat a memóriában. Visszatérés „RETURN"-nel. A képernyő bal felső sarkában nyomon követhetjük az éppen aktuális sorszámot. Az „F1" billentyű lenyomásával gyors lejátszás üzembe léphetünk.

KIMENTÉS

Ha van a memóriában zene, úgy a megszerkesztett zene adatait szalagra vagy lemezre menthetjük ki. A program csak a zene adatait menti ki, ezek az adatok tehát csak a zeneszerkesztő program számára érthetőek. A kimentett adatok címe elé egy megkülönböztetés kerül, amelyről felismeri a program a zeneadatokat. Ez a megkülönböztetés a lemez-tartalomjegyzékben is jól láthatóan elkülöníti a zeneadatokat.

BETÖLTÉS

A kimentett zeneadatok visszatöltése.

ÖSSZEKAPCSOLÁS

A kész zenét egy meglévő programhoz kapcsolhatjuk, vagy a zenéhez tetszőleges programot írhatunk. Ehhez tudni kell a programvég mutatók értékét. Példa egy meglévő programmal való összekapcsolására:
Betöltjük a programot, megjegyezzük a 45 és 46-os cím tartalmát. Ezután betöltjük a zeneszerkesztő programot és a zeneadatokat. Összekapcsolás üzemmódot kérünk. Beírjuk a kért adatokat és feljegyezzük a közölt adatokat. A „RETURN" megnyomása után betöltjük az összekapcsolni kívánt programot és beírjuk a 45 és 46 címekre az adatokat. Ezzel a két programot összekapcsoltuk. Ha a programon változtatunk az összekapcsolás után, akkor a zenét csak a programvég mutatókból számított „SYS" utasítással hívhatjuk.
SYS 256*PEEK(46)+PEEK(45)–
– a zene hossza

Figyelem!
Ha összekapcsolás üzemmóddal elhelyeztünk egy zeneprogramot a gép memóriájában, akkor ezt a zenét már nem lehet egyszerűen megváltoztatni. Célszerű tehát minden elkészült zenét először kimenteni!
Soha ne változtassunk a főprogramon akkor, ha a zene nincs leállítva!

HELP

A „HELP" funkció segítséget nyújt a legfontosabb billentyűk és a címek között való eligazodásban.

A ZENEPROGRAM MŰKÖDÉSE

Ha valaki tegnap ült életében először számítógép előtt, és nem érti a most következő sorokat, az azért van, mert ezek inkább haladóknak íródtak!
A zeneprogramot a **hardver megszakító rutin** működteti. Ez teszi lehetővé azt, hogy a gép zene közben teljesen szabad a főprogram számára. Ha valaki a hardver megszakító rutint használni akarja, akkor a megszakítás vektort a zene hívása előtt írja át. A zeneprogram eltárolja a régi vektort és dolga végeztével erre adódik át a vezérlés.
A zeneprogram használ néhány **nullalapos címet.** Emiatt zene közben **nem használható** a magnó és az RS 232-es csatorna.

A **kontroll regisztereket** nem lehet a SID regisztereibe írt számmal módosítani, mert a nullalapon eltárolt adatok minden hang megszólalásakor visszaíródnak. A hullámalakokat tehát mindig a megadott nullalapos címen módosítsuk!

```
30D4  78        SEI
30D5  A9 60     LDA #60       ;RTS kódja.
30D7  85 A5     STA   A5
30D9  20 A5 00  JSR 00A5
30DC  BA        TSX           ;A veremből vissza a címet.
30DD  BD 00 01  LDA 0100,X
30E0  85 A6     STA   A6
30E2  CA        DEX
30E3  BD 00 01  LDA 0100,X
30E6  85 A5     STA   A5
30E8  A9 AB     LDA #AB       ;Új megszakítás vektor
30EA  18        CLC           ;kiszámítása.
30EB  65 A5     ADC   A5
30ED  90 02     BCC 30F1
30EF  E6 A6     INC   A6
30F1  85 A5     STA   A5
30F3  AD 14 03  LDA 0314      ;Régi megszakitás vektor
30F6  85 F7     STA   F7      ;elmentése.
30F8  AD 15 03  LDA 0315
30FB  85 F8     STA   F8
30FD  A5 A6     LDA   A6      ;Új megszakitás vektor
30FF  8D 15 03  STA 0315      ;betöltése.
3102  A5 A5     LDA   A5
3104  8D 14 03  STA 0314
3107  18        CLC           ;Hangjegytáblázat címének
3108  69 4D     ADC #4D       ;kiszámitása
310A  90 02     BCC 310E
310C  E6 A6     INC   A6
310E  85 B4     STA   B4      ;és betöltése.
3110  E6 A6     INC   A6
3112  85 A5     STA   A5
3114  A5 A6     LDA   A6
3116  85 B5     STA   B5
3118  A9 18     LDA #18       ;Zeneadatok mutatójának
311A  18        CLC
311B  65 A5     ADC   A5      ;kiszámitása
311D  90 02     BCC 3121
311F  E6 A6     INC   A6
3121  85 A5     STA   A5      ;és betöltése.
3123  A0 00     LDY #00
3125  B1 A5     LDA  (A5),Y   ;Az első zeneadat
3127  85 A7     STA   A7      ;a szólamszám.
3129  C8        INY
312A  A2 00     LDX #00
312C  8A        TXA
312D  48        PHA
312E  F0 08     BEQ 3138
3130  A9 00     LDA #00
3132  18        CLC
3133  69 07     ADC #07
3135  CA        DEX
3136  D0 FA     BNE 3132
3138  AA        TAX
3139  B1 A5     LDA  (A5),Y   ;Az ADSR értékek:
313B  9D 05 D4  STA D405,X    ;Felfutás/lecsengés,
313E  C8        INY
313F  B1 A5     LDA  (A5),Y
3141  9D 06 D4  STA D406,X    ;tartás/elengedés.
3144  C8        INY
3145  B1 A5     LDA  (A5),Y
3147  95 AA     STA   AA,X    ;Hullámalak.
3149  68        PLA
314A  AA        TAX
314B  C8        INY
314C  E8        INX
314D  E4 A7     CPX   A7      ;Minden csatorna kész?
314F  D0 DB     BNE 312C
3151  B1 A5     LDA  (A5),Y   ;Igen.
3153  8D 18 D4  STA D418      ;Hangerő.
3156  C8        INY
3157  B1 A5     LDA  (A5),Y
3159  85 A9     STA   A9      ;Tempó.
315B  C8        INY
315C  98        TYA
315D  18        CLC
315E  65 A5     ADC   A5      ;Az aktuális hang mutatójának
3160  90 02     BCC 3164
3162  E6 A6     INC   A6      ;kiszámitása
3164  38        SEC
```

```
3165  E5 B4     SBC  B4
3167  85 A5     STA  A5
3169  A5 A6     LDA  A6
316B  E5 B5     SBC  B5
316D  85 A6     STA  A6
316F  A5 95     LDA  A5
3171  38        SEC
3172  E9 18     SBC #18
3174  B0 02     BCS 3178
3176  C6 A6     DEC  A6
3178  85 A5     STA  A5        ;és betöltése.
317A  A9 01     LDA #01        ;Az ugrást engedélyező
317C  85 AE     STA  AE        ;regiszterbe 1-et tölt.
317E  A9 02     LDA #02        ;2-t tölt az
3180  85 AB     STA  AB        ;időszámláló regiszterbe
3182  85 B0     STA  B0        ;és a stop regiszterbe.
3184  58        CLI
3185  60        RTS            ;Visszatérés.
3186  A0 00     LDY #00
3188  A5 B0     LDA  B0        ;Mi van a stop regiszterben?
318A  F0 47     BEQ 31D3       ;Ha 0 akkor vége,
318C  C9 01     CMP #01        ;ha 1 akkor szünet.
318E  D0 16     BNE 31A6       ;Egyik sem.
                                              SZÜNET
3190  A5 A7     LDA  A7
3192  C9 01     CMP #01        ;1 szólam?
3194  F0 0A     BEQ 31A0
3196  C9 02     CMP #02        ;2 szólam?
3198  F0 03     BEQ 319D
319A  8C 12 D4  STY D412       ;3.csatorna leáll.
319D  8C 0B D4  STY D40B       ;2.csatorna leáll.
31A0  8C 04 D4  STY D404       ;1.csatorna leáll.
31A3  6C F7 00  JMP(00F7)      ;Hardver megszakító rutinra
31A6  C6 AB     DEC  AB        ;Időszámláló csökkentése.
31A8  D0 F9     BNE 31A3       ;0 lett?
31AA  A5 B4     LDA  B4        ;Igen: Hangváltás lesz.
31AC  18        CLC
31AD  65 A5     ADC  A5        ;A következő hangadat mutató
31AF  85 A5     STA  A5
31B1  A5 B5     LDA  B5
31B3  65 A6     ADC  A6        ;kiszámítása
31B5  85 A6     STA  A6
31B7  A9 18     LDA #18
31B9  18        CLC
31BA  65 A5     ADC  A5
31BC  90 02     BCC 31C0
31BE  E6 A6     INC  A6
31C0  85 A5     STA  A5        ;és betöltése.
31C2  B1 A5     LDA  (A5),Y
31C4  C9 FF     CMP #FF        ;Végjel?
31C6  F0 0B     BEQ 31D3
31C8  C9 80     CMP #80        ;Ugrás?
31CA  F0 29     BEQ 31F5
31CC  C9 90     CMP #90        ;Memóriába írás?
31CE  F0 42     BEQ 3212
31D0  18        CLC
31D1  90 58     BCC 322B       ;FELTETEL NELKÜLI UGRAS.
                                          VÉGE A DALNAK
31D3  A0 00     LDY #00        ;
31D5  A5 A7     LDA  A7
31D7  C9 01     CMP #01        ;1 szólam?
31D9  F0 0A     BEQ 31E5
31DB  C9 02     CMP #02        ;2 szólam?
31DD  F0 03     BEQ 31E2
31DF  8C 12 D4  STY D412       ;3. szólam leáll.
31E2  8C 0B D4  STY D40B       ;2. szólam leáll.
31E5  8C 04 D4  STY D404       ;1. szólam leáll.
31E8  A5 F7     LDA  F7        ;A régi megszakítás vektor
31EA  8D 14 03  STA 0314       ;betöltése
31ED  A5 F8     LDA  F8
31EF  8D 15 03  STA 0315       ;és visszaírása.
31F2  6C F7 00  JMP(00F7)      ;Hardver megszakító rutinra
                                                  UGRAS
31F5  A5 AE     LDA  AE        ;Mi van az ugrást
31F7  D0 06     BNE 31FF       ;engedélyező regiszterben?
31F9  C8        INY            ;0. tehát nincs egedélyezve.
31FA  C8        INY            ;3 byte-ot figyelmen kívül
31FB  C8        INY            ;kell hagyni.
31FC  18        CLC
31FD  90 C3     BCC 31C2       ;FELTETEL NELKÜLI UGRAS.
31FF  C6 AE     DEC  AE        ;Ugrást engedélyező regiszter
3201  C8        INY
3202  B1 A5     LDA  (A5),Y    ;Az új mutató kiolvasása
3204  48        PHA
3205  C8        INY
3206  B1 A5     LDA  (A5),Y
3208  85 A6     STA  A6
320A  68        PLA
320B  85 A5     STA  A5        ;és betöltése.
320D  A0 00     LDY #00
320F  18        CLC
3210  90 98     BCC 31AA       ;FELTETEL NELKÜLI UGRAS.
                                         MEMÓRIABA ÍRAS
3212  C8        INY
3213  B1 A5     LDA  (A5),Y    ;A cím alsó byte-ja.
3215  85 F9     STA  F9
3217  C8        INY
3218  B1 A5     LDA  (A5),Y    ;A cím felső byte-ja.
321A  85 FA     STA  FA
321C  C8        INY
321D  98        TYA
321E  48        PHA
321F  B1 A5     LDA  (A5),Y    ;A beírandó adat.
3221  A0 00     LDY #00
3223  91 F9     STA  (F9),Y    ;Beírás.
3225  68        PLA
3226  A8        TAY
3227  C8        INY
3228  18        CLC
3229  90 97     BCC 31C2       ;FELTETEL NELKÜLI UGRAS.
                                            ZENÉLNI KEL
322B  A2 00     LDX #00
322D  B1 A5     LDA  (A5),Y    ;A következő hangadat.
322F  C9 A0     CMP #A0        ;Változatlan hang?
3231  F0 5F     BEQ 3292
3233  98        TYA            ;Nem.
3234  48        PHA
3235  8A        TXA
3236  48        PHA
3237  F0 09     BEQ 3242
3239  A9 00     LDA #00
323B  18        CLC
323C  69 07     ADC #07
323E  CA        DEX
323F  D0 FA     BNE 323B
3241  AA        TAX
3242  48        PHA
3243  B5 AA     LDA  AA,X      ;A kontroll regiszter.
3245  29 FE     AND #FE        ;Kapubit kikapcsolása.
3247  9D 04 D4  STA D404,X
324A  B1 A5     LDA  (A5),Y
324C  D0 06     BNE 3254       ;Szünet?
324E  85 F9     STA  F9        ;Igen.
3250  85 FA     STA  FA
3252  F0 29     BEQ 327D
3254  C9 B0     CMP #B0        ;Kapu bit ki?
3256  D0 06     BNE 325E
3258  68        PLA
3259  18        CLC
325A  90 32     BCC 328E       ;FELTETEL NELKÜLI UGRAS.
                                     EGY BYTE SZÉTVALASZTASA
                                         OKTÁVRA ÉS HANGRA
325C  EA        NOP
325D  EA        NOP
325E  4A        LSR            ;Az oktáv leválasztása
325F  4A        LSR
3260  4A        LSR
3261  4A        LSR
3262  AA        TAX            ;és betöltése X-be.
3263  BA A5     LDA  (A5),Y
3265  29 0F     AND #0F        ;A hang mutató leválasztása
3267  0A        ASL
3268  A8        TAY            ;és betöltése Y-ba.
3269  B1 B4     LDA  (B4),Y    ;A frekvencia konstans
326B  85 F9     STA  F9        ;kikeresése a táblázatból
326D  C8        INY            ;7. oktávra,
326E  B1 B4     LDA  (B4),Y
3270  85 FA     STA  FA        ;és betöltés nullalapra.
3272  E0 07     CPX #07
3274  F0 07     BEQ 327D
3276  46 FA     LSR  FA        ;Visszaosztás a kért
3278  66 F9     ROR  F9        ;oktávra.
327A  E8        INX
327B  D0 F5     BNE 3272
327D  68        PLA
327E  AA        TAX
327F  A5 F9     LDA  F9        ;A frekvencia konstans
3281  9D 00 D4  STA D400,X     ;betöltése a SID-be.
3284  A5 FA     LDA  FA
3286  9D 01 D4  STA D401,X
3289  B5 AA     LDA  AA,X      ;A kapubit bekapcsolása.
328B  9D 04 D4  STA D404,X
328E  68        PLA
328F  AA        TAX
3290  68        PLA
3291  A8        TAY
3292  C8        INY
3293  E8        INX
3294  E4 A7     CPX  A7        ;Minden csatorna kész?
3296  D0 95     BNE 322D
                                          IDŐ KISZAMÍTASA
3298  B1 A5     LDA  (A5),Y
329A  D0 06     BNE 32A2       ;0 ütemhossznál
329C  A9 01     LDA #01        ;1-et irunk
329E  85 AB     STA  AB        ;az időszámláló regiszterbe.
32A0  D0 0D     BNE 32AF
32A2  A6 A9     LDX  A9        ;Tempó az X-be.
32A4  CA        DEX
32A5  F0 06     BEQ 32AD
32A7  18        CLC            ;A tempó és az ütemhossz
32A8  71 A5     ADC  (A5),Y    ;összeszorzása.
32AA  18        CLC
32AB  90 F7     BCC 32A4
32AD  85 AB     STA  AB        ;Eredmény az időszámlálóba.
32AF  C8        INY
32B0  98        TYA            ;A következő hangadat
32B1  18        CLC            ;mutatójának kiszámítása
32B2  65 A5     ADC  A5
32B4  90 02     BCC 32B8
32B6  E6 A6     INC  A6
32B8  38        SEC
32B9  E5 B4     SBC  B4
32BB  85 A5     STA  A5
32BD  A5 A6     LDA  A6
32BF  E5 B5     SBC  B5
32C1  85 A6     STA  A6        ;és betöltése.
32C3  A5 A5     LDA  A5
32C5  38        SEC
32C6  E9 18     SBC #18
32C8  B0 02     BCS 32CC
32CA  C6 A6     DEC  A6
32CC  85 A5     STA  A5
32CE  6C F7 00  JMP(00F7)      ;Hardver megszakító rutinra.
```

A használt nullalapos címek:

Hex.	Dec.	
A5–A6	165–166	– a következő hang mutatója
A7	167	– szólam száma
A9	169	– tempó
AA	170	– C.R 1
AB	171	– idő számláló
AE	174	– ugrást engedélyező regiszter
B0	176	– STOP
B1	177	– C.R 2
B4–B5	180–181	– hangfrekvencia táblázat mutató
B8	184	– C.R 3
F7–F8	247–248	– a régi megszakítás vektor

A zene tempója programból is módosítható, de ügyeljünk arra, hogy a tempó értéke 1 és 15 között legyen!

A **„STOP" regiszterbe** 0-át írva a zene megáll, az eltárolt megszakítás vektor visszaírásra kerül. Ha a „STOP" regiszterbe 1 kerül, akkor a zene megáll, de a megszakítás vektor nem íródik vissza. Ilyenkor a program figyeli a „STOP" regisztert, és ha ennek tartalma 0-tól vagy 1-től különböző lesz, akkor a zene folytatódik.

A **171**-es címen helyezkedik el az **időszámláló regiszter.** Ha a zene alatt ennek tartalma 1 lesz, akkor következik be a hangváltás. A főprogramot szinkronizálhatjuk a zenéhez, ha figyeljük az időszámláló regisztert, és a szükséges változtatásokat akkor ejtjük meg, amikor hangváltás van. A **memóriába írás** műveletekkel viszont a zeneprogramból „nyúlhatunk bele" a főprogramba, vagy akár a grafikába is.

Azoknak az olvasóinknak, akik mélyebben szeretnének a program lelkivilágába látni, közöljük annak a programrészletnek a gépi nyelvű listáját, amelyik a tulajdonképpeni „zenélő". A program működésének megértéséhez ismerni kell a zeneadatok kódolási módszerét. A zeneadatok sorrendje szigorúan kötött. Az első adat mindig a szólamszám, majd következnek az ADSR értékek és a hullámalak szólamonként. A következő adat a hangerő, majd a tempó. Ezután a hangadatok vannak a következők szerint:

Az oktáv és a hang egyetlen byteban van elhelyezve. Az oktáv a felső, a hang pedig az alsó 4 bit-ben. Ahány szólam van kiválasztva, annyi hangadatnak, vagy azt helyettesítő (pl. szünet) adatnak kell egymás után lenni. A $80 és annál nagyobb számok mind vezérlő kódok. A megfelelő számú hangadatot követi az ütemhosszúság.

A zeneadatok végét egy $FF karakter jelzi.

Kővári László

A következő három kis rutin bármelyik Commodore-on működik. Igazán semmi különleges nincs bennük, de mégis hasznos segédeszközei lehetnek a programfejlesztőnek. Úgy gondoljuk, hogy ilyesfajta picike rutinjaikat nyugodtan beküldhetik olvasóink. Hiszen a programfejlesztőnek úgyis van egy lemeze, kazettája, amelyen semmi más nincs, mint azok a segédprogramocskák, amelyekre bármikor szüksége lehet. S ezeken a lemezeken mindig van még egy kis hely – újabb hasznos „cédulák"-nak.

LÁNC-LÁNC

Ez a program a lemezen lévő blokkok láncolásának meghatározására szolgál. Egy blokk track ill. sector számát megadva a program kiírja az egymás után következő blokkok paramétereit: track és sector számaikat.

```
1 REM ********************************
2 REM *    C=UJSAG                    *
3 REM *    LANC-LANC                  *
4 REM *    PROGRAM: ZSOLDOS LASZLO    *
5 REM ********************************
6 REM
10 OPEN2,8,2,"#":OPEN15,8,15
20 INPUT"TRACK,SECTOR";T,S
30 PRINT#15,"U1 2 0";T;S:PRINTT;S
40 GET#2,X$,Y$:IF X$="" THEN X$=CHR$(0)
50 IF Y$="" THEN Y$=CHR$(0)
60 T=ASC(X$):S=ASC(Y$)
70 IF T=0 THEN GOTO 20
80 GOTO 30
```

HASONLÍTÓ

A lemezen található két kijelölt file összehasonliását végzi el, az első eltérésnél leáll és kijelzi az eltérő byte sorszámát.

```
1 REM ********************************
2 REM *    C=UJSAG                    *
3 REM *    HASONLITO                  *
4 REM *    PROGRAM: ZSOLDOS LASZLO    *
5 REM ********************************
6 REM
10 INPUT "FILE 1";F1$
20 INPUT "FILE 2";F2$
30 OPEN2,8,2,F1$
40 OPEN3,8,3,F2$
50 I=1
60 GET#2,A$:GET#3,B$:IF A$=B$ AND ST=0 THEN I=I+1:GOTO60
70 IF ST=0 THEN PRINT"ELTERES A"I". BYTE-NAL"
80 CLOSE2:CLOSE3
```

KIÍRÓ

Egy tetszőleges, a lemezen található file képernyőre történő kiírását végzi el. Jól használható lemezen lévő szekvenciális és szöveges program file-ok ellenőrzésére is.

```
1 REM ********************************
2 REM *    C=UJSAG                    *
3 REM *    KIIRO                      *
4 REM *    PROGRAM: ZSOLDOS LASZLO    *
5 REM ********************************
6 REM
10 INPUT"FILE";F$
20 OPEN 2,8,2,F$
30 GET#2,A$:IF ST=0 THEN PRINTA$;:GOTO30
40 CLOSE2
```

KERESŐ

Ennek a pici programnak a segítségével C 16, ill. Plus/4-es gépen lehetőség nyílik a program „meghallgatására" a tv hangszórója keresztül. A programot beírjuk a gépbe, majd RUN-nal futtatjuk. Tegyünk egy olyan kazettát a datasettbe, amit meg akarunk hallgatni. A gép kérdésére nyomjuk meg az „I" gombot, majd a magnót állítsuk lejátszásra. RESET gomb megnyomása esetén SYS 8192-vel aktivizálhatjuk újra a programot.

```
1 REM ********************************
2 REM *    C= UJSAG                   *
3 REM *    KERESO                     *
4 REM *    PROGRAM: SZABOLCSI SZABOLCS *
5 REM ********************************
6 REM
10 FOR X=8192 TO 8212:READ A$:POKE X,DEC(A$)
20 NEXT X
30 DATA A9,00,AC,06,FF,8D,06,FF
40 DATA A5,01,69,38,8D,11,FF,8D
50 DATA 19,FF,4C,00,20
60 SCNCLR
70 PRINT"KEZDHETJÜK (I/N)?":GETKEY A$
80 IF A$="I" THEN SYS 8192:ELSE 80
```

A program jobb megértéséhez a gépi kódú listát is közöljük.

```
. 2000   A9 00      LDA #$00
. 2002   AC 06 FF   LDY $FF06
. 2005   8D 06 FF   STA $FF06
. 2008   A5 01      LDA $01
. 200A   69 38      ADC #$38
. 200C   8D 11 FF   STA $FF11
. 200F   8D 19 FF   STA $FF19
. 2012   4C 00 20   JMP $2000
```

PRINT SHOP és PRINTMASTER

Ezek azok a programok, amelyeket valamennyien ismerünk, csak épp nem biztos, hogy tudunk róluk. A helyzet ugyanis az, hogy a város tele van csúfnál csúfabb számítógépes plakátokkal, amelyek ezekkel a programokkal készülnek. A plakátok csúfságáról persze nem a programok tehetnek, hiszen azokat nem azért készítették, hogy a plakátművészettel vetelkedjenek. A maguk helyén és a maguk módján ezek a programok jó szolgálatot tehetnek.



1. ÁBRA

A PROGRAM SZÜLETÉSE

A **Print Shop** Steve Wozniak, az Apple cég alapító tulajdonosának saját kívánsága volt, és ebből következik, hogy a program először az Apple II sorozat gépein futott. De a programban nemcsak Mr. Wozniak lelte örömét, hanem hamarosan Amerika és Európa Apple tulajdonosai. A program kirobbanó üzleti sikerén felbuzdulva a programot készítő háromfős csoportból kiválva Corey Kosak és Martin Kahn elkészítették a rendszer Commodore 64-en és az Atari XL/XE gépeken futó változatát. A Print Shop alig egy évig volt egyeduralkodó a piacon, amikor – ahogy az lenni szokott – a konkurencia megjelent egy hasonló termékkel. A Berkeley Egyetem programozói és kiadó vállalatuk a Berkeley Softworks hatalmas reklámhadjárat kíséretében bejelentette a **PRINT MASTER** című programot 1985-ben. Mindkét programnak megjelent azóta egy újabb változata is, a **Print Shop The Companion** és a **Print Master Plus** (talán egy későbbi számban bemutatásra kerülnek ezek a programok is). A Print Shop-hoz megjelent még három lemez, ami különféle grafikai kiegészítéseket tartalmaz. (Erről még később szó lesz.) Mindkét program a mai napig is kapható (habár a Print Shop-ból jóval többet adnak el) az amerikai kontinensen és az óvilágban is.

A RENDSZER

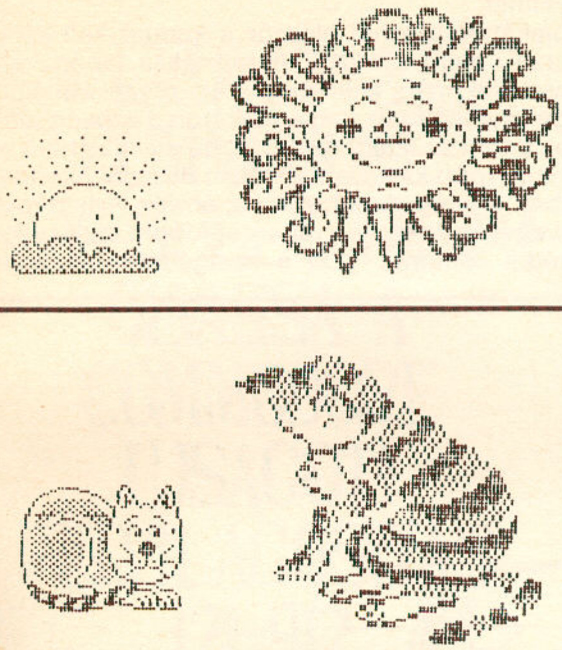
Látván a Print Shop kiváló kezelési rendszerét, a Print Master alkotói azon mit sem változtattak. A program indulása után egy **menü** jelentkezik be, ahonnan az **alprogramok hívhatók** a cursor nyilak vagy egy joystick esetleg Koala Pad és a menüpontokat állandóan inverzbe átváltó mező segítségével. A program összes kérdésére ezzel a módszerrel lehet válaszolni. Minden kérdésről vissza lehet lépni az előzőre egy gombnyomással. Ha elégszer nyomjuk az adott billentyűt, a főmenüig érhetünk vissza.
A Print Shop nagy hátránya az, hogy az alaprogramokat mindig újra betölti.

A LEHETŐSÉGEK:

Mindkét program abban az országban született meg, ahol az emberek születésnapi, karácsonyi és egyéb üdvözlő kártyák tíz- és százezreiben dúskálhatnak, ahol a legegyszerűbb átlag polgárnak is van saját nevével és címével ellátott levélpapírja és borítékja, és ahol a nyomdai szolgáltatások igénybevétele ugyanolyan természetes, mint a hétvégi bevásárlás.
Talán már nincs is szükség, hogy a két program lehetőségeit leírjam, csak a pontosság kedvéért teszem meg. Mindkét rendszerrel lehet tervezni és kivitelezni **üdvözlő kártyákat, plakátokat, falragaszokat, transzparenseket, fejléces levélpapírt,** és a Print Master még **naptárt** is tud készíteni.

AZ ÁLLANDÓ GRAFIKAI ELEMEK

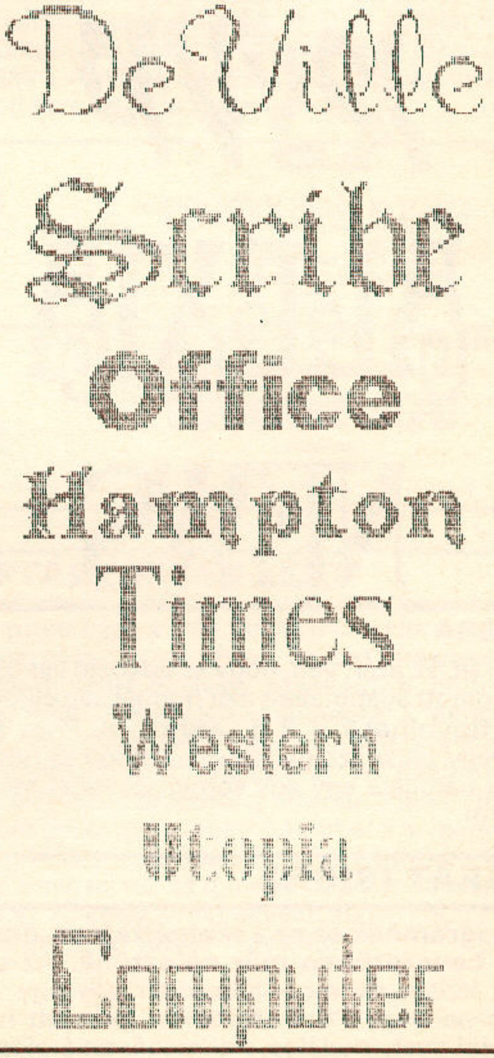
A tervezéshez különféle előre elkészített apró rajzok, szimbólumok, különféle keretek és betűtípusok állnak rendelkezésünkre.
A grafikai elemeket, azaz a rajzokat fel lehet használni minden üzemmódban. Ha akarjuk, ugyanazt a szívet megjeleníthetjük egy üdvözlő kártyán és egy plakáton is. A lemezen találhatók meg



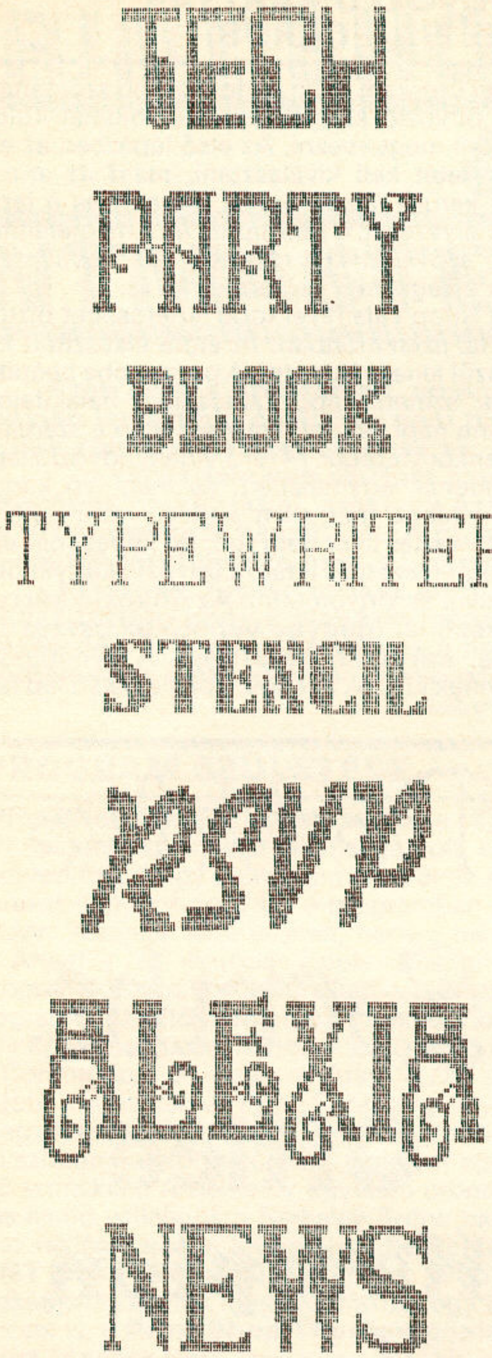
ezeknek a rajzoknak az adatai és a kért ábra bármikor behívható. A Print Master nagyjából ugyanazokat a rajzokat használja fel, amelyeket a Print Shop, csak valamivel nagyobb felbontásban és valósághűbb, olykor kissé giccsesebb kivitelben. **Az 1. ábrán a Print Shop és a Print Master azonos jellegű apró rajzainak egy részét mutatjuk be.** (A különbség szemmel látható.) A Print Master nemcsak azokat az elemeket tartalmazza, amelyeket idősebb testvére, hanem több mást is. Eredetileg a Print Shop alapkiépítésében 60, a Print Master-ében 122 db rajz található. A Print Shop-hoz külön meg lehet vásárolni még rajzokat. Az Egyesült Államokban 15–18 dollár körüli áron adják a Print Shop Graphic Library című lemezeket. Júniusig négy ilyen lemez van forgalomban. A negyedik címe: Print Shop Graphics Library Holiday Edition. Minden egyes lemez 120 ábrát tartalmaz. Aki viszont nem talál magának megfelelőt egyiken sem, az készíthet magának is a beépített grafikai szerkesztőrutin segítségével. (Részletesen később.) **A különféle szövegek felírásához mindkét rendszerben 8-féle betűtípus közül lehet válogatni. A 2. ábra a programok betűtípusait mutatja be.** A Print Master nagy előnye, hogy egy adott munkán lehet többféle betűt is használni, ellentétben a Print Shop-pal.
A különböző szöveg – rajz kombinációkat keretek közé tudjuk zárni. **A Print Master 11-féle, a Print Shop 9-féle keretet ismer.**
Azt hibának tartom mindkét program esetében, hogy keretet és betűtípust a felhasználó nem tervezhet, s ezáltal ékezetes betűket sem tudunk használni. (Ebből is látszik, hogy az amerikai szoftverkiadóknak nem Magyarország vagy Csehszlovákia a fő piacuk.)

AZ ÜDVÖZLŐ KÁRTYÁK (GREETING CARDS)

Mindkét program talán legfontosabb szolgáltatása. Az üdvözlő kártyák mindkét oldalát különkülön kell megtervezni. Az első lépésben az első oldal keretét kell kiválasztani, majd az arra az oldalra kerülő rajzot, annak méretét és a lapon való elhelyezését, a betűtípust és a megjelenítésre váró szöveget kell megadni. És ugyanezt az eljárást elvégezni a belső oldallal is.
A Print Shop valamivel több lehetőséget biztosít ebben az üzemmódban. Itt előre elkészített kártyák közül lehet válogatni. A programba beépítettek egy születésnapi, karácsonyi, hálakifejező, Valentine napi (angol nyelvterületen a szerelmesek ilyenkor (február 14-én) vidám üdvözlő lapokat küldenek egymásnak), és meghívó kártyák rajzát, amihez már nem kell tervezni semmit. A Print Master ilyet nem tud, de képes már egyszer megtervezett és letárolt üdvözlő kártyát diskről behívni és kinyomtatni. **Az üdvözlő kártyák tervezésénél igénybe vehető az összes rajzocska, keret és betűtípus.** A nyomtatóból úgy jönnek ki ezek a lapok, hogy a belső lap fejjel



2. ÁBRA



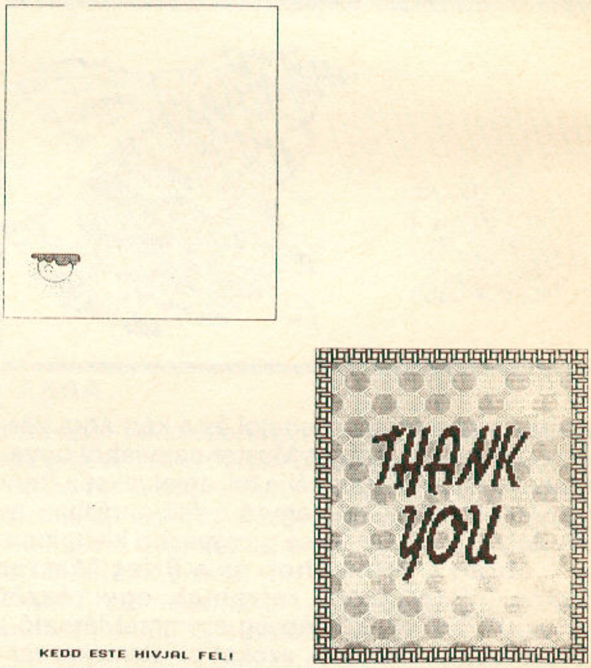
2. ÁBRA

lefelé áll. Ez azért van, hogy az üdvözlő lap összehajtogatott állapotában már normálisan álljon. **Ez megfigyelhető a 3–4. ábrán.** A Print Shop még egy plusz szolgáltatása az, hogy az üdvözlő lapok hátuljára egy egy sornyi szöveget még lehet írni.

PLAKÁT (SIGN)

A programnak ez az a szolgáltatása, amelyre a bevezetőben már céloztunk. Az utcán járva lépten-nyomon felbukkan egy-egy Print Shop-pal vagy Print Master-rel tervezett hirdetés. Hol nyelviskolákra, hol magántanárokra, hol pedig ötödosztályú rockkoncertekre hívják fel a figyelmet.

A plakáttervezésnél először a keretet kell kiválasztani, majd a rajzot. A rajznak a lapon való beosztását meg kell adni. Van olyan variáció, hogy csak egy rajz van a plakáton a szöveg alatt és az kitölti az egész területet, de elképzelhető az is, hogy több kicsi van a képen, de saját elrendezésben is megjeleníthetők az ábrák. A begépelt szöveget kívánságra a rendszer jobbra, balra vagy középre tömöríti. (Ezt a szolgáltatást minden





3–4. ÁBRA

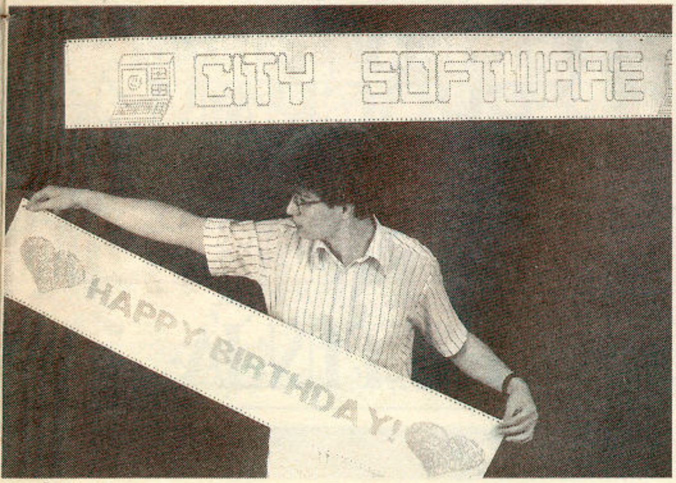


5. ÁBRA

üzemmódban nyújtja a program.) A plakátokat A4-es írógéppapír méretben nyomtatja ki a gép. **A Print Master-rel készült plakátot mutat be az 5. ábra.**

TRANSZPARENSEK (BANNER)

Az egyik leglátványosabb szolgáltatása mindkét rendszernek. **Adott szövegeket akár több méter hosszúságban lehet megjeleníteni.** A Print Shop-nak ebben a kategóriában is valamivel jobbak a paraméterei. Ugyanabból a betűtípusból, azonos karakterekből álló sorozatból a Print Shop 56-ot, a Print Master csak 37 jelet tudott elfogadni. A Print Master megkérdezi, hogy milyen rajzzal nyomtassa ki a szöveget. Ezt a rajzot a szöveg elé és mögé teszi. A Print Shop



6. ÁBRA

annyival intelligensebb, hogy megkérdezi, hogy elé, mögé vagy mindkét végére rakja-e ki az ábrát.
A 6. ábra mutatja be a két programnak ezt a szolgáltatását.

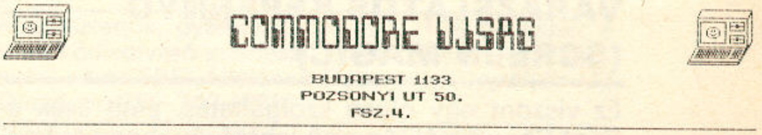
LEVÉLPAPÍR (LETTERHEAD)

E rutin segítségével bárki szert tehet névre szóló levélpapírra. A papír alsó részét és felső és alsó 1/6 részét kell megtervezni. Mindkét helyen lehet egy nagybetűs szöveg és több apró, valamint két rajz. De itt is elképzelhető, hogy ne, vagy csak egy rajz legyen. Itt is lehet jobbra, balra és középre tömöríteni. A levélpapír készítésben minden tekintetben a Print Shop viszi el a pálmát. Azt hiszem, felesleges szószaporítás helyett **beszéljen a 7. és a 8. ábra.**

RAJZSZERKESZTŐ (GRAPHIC EDITOR)

Ha a kis rajzok nem elégítik ki valakinek az igényeit teljes mértékben, az **tervezhet saját ábrákat** is a Graphic Editor rutin segítségével. A Print Master ebben a szolgáltatásban jóval többet nyújt. **A kép felbontása 87 × 51** pont, míg a Print Shop-é csak **43 × 44.** A Print Master képes függőleges és vízszintes irányban is tükrözni és invertálni. A rajzok mindkét programban felvehetők lemezre és ki is nyomtathatók.

PRINT SHOP



PRINTMASTER



7–8. ÁBRA

NAPTÁR (CALENDER)

Ezt a rutint csak a Print Master tulajdonosai tudják használni. **A gép havi és heti naptárokat tud csak szerkeszteni.** Megkérdezi, hogy melyik év melyik hónapjáról kell naptár a felhasználónak, és a programba épített öröknaptár algoritmus segítségével megtervezi a naptárunkat. A hónap nevét eredetileg angolul írja ki, de egy gombnyomás segítségével át lehet írni magyarra. A kalendáriumunk tetejére el lehet helyezni egy tetszőleges rajzot, és minden naphoz be lehet írni egy alig 15 karakternyi emlékeztetőt, ahogy ezt a **9. ábrán meg lehet figyelni.**

Junius 1987

SUN	MON	TUE	WED	THU	FRI	SAT
	1	2 Angol vizsga	3 Angol vizsga	4	5	6
7	8	9	10	11	12	13
14	15 Vizsgak	16	17	18 Ban-kett	19 Szent-endre	20
21	22	23 Szent-endre	24	25 Cseh-szlo-vakia	26	27
28 Lengyel orszag	29	30				

9. ÁBRA

VARÁZSLATOS KÉPERNYŐ (SCREEN MAGIC)

Ez viszont egy olyan szolgáltatás, amit csak a Print Shop kínál. Az első lehetőség, hogy ebben az üzemmódban használhatjuk a legnagyobb méretű karaktereket. Készíthetünk plakátot is, de a felhasználható kereteket, rajzokat itt nem tudjuk elérni. **A Screen Magic legizgalmasabb szolgáltatása: a kaleidoszkóp.** A képernyőre véletlenszerű szimmetrikus geometriai ábrák helyezhetők el, ezeket a gép generálja, majd ezekre ráviszi a kívánt szöveget. Mindezt természetesen ki is tudjuk nyomtatni. **A 10. ábra ezzel az alprogrammal készült.** Ennek a rutinnak egy másik üzemmódjában tetszőleges nagyfelbontású rajzot is ki tudunk nyomtatni. Az eljárás menete a következő: a nyomtatásra váró rajzot a 16384–24576 közti memória címekre kell tölteni, majd felvenni lemezre. Ezután a Print Shop Screen Magic rutinjába betölteni és onnan már csak a PRINT parancsot kell megadni a gépnek.

VÉGSŐ ÖSSZEHASONLÍTÁS

Nem jelenthetjük minden lelkiismeret-furdalás nélkül, hogy az egyik program jobb, mint a másik. **Van amiben a Print Shop tud többet, van amiben a Print Master.** A Print Shop nagy hiányossága a Print Masterrel szemben, hogy csak a Commodore grafikus nyomtatókat kezeli. A Print Masterre rá lehet kapcsolni a következő nyomtatókat: OKIMATE 10/20, EPSON FX/RX, STAR SG–10/15, OKI 92, OKI 93, C–Itoh 8510. Apróság az is, hogy a Print Masteren a képernyő színét is beállíthatja a tulajdonos és az is, hogy a Print Masternek a karcsú keretvonalaknak és a kisbetű és nagy betű használatának köszönhetően jóval kellemesebb a megjelenése. Módfelett kiváló az a tulajdonsága is, hogy a kinyomtatásra váró ábrát a képernyőn is megmutatja és nem pár méter hasznos festékszalag elkoptatása után kell először találkozni alkotásunkkal. A Print Shopnak nagy előnye, hogy kevesebbet tölt lemezről mint társa, rövidebb ideig gondolkodik, és hogy minden egyes nyomtatásnál meg lehet adnunk, hogy hány példányt szeretnénk az adott termékből.

A PROGRAM BESZERZÉSE

A Print Shop és a Print Master megvásárlása Ausztriában vagy a Német Szövetségi Köztársaságban némi utánjárást igényel. **A legnagyobb szoftverkereskedők már egyiket sem tartják.** A Print Master Plust májusban még lehetett kapni az Interfunk kölni boltjában 198 márkáért. Az Amerikai Egyesült Államokban és Kanadában nem ütközik különösebb nehézségekbe a programok beszerzése. Mindkét program kapható az összes közismert kereskedőnél 25–35 dollár közti áron. De én úgy vélem, hogy a szűkös valutakeretből nem érdemes költeni ezekre a programokra, mert mind a kettő jól ismert hazai szoftver kalóz berkekben.

SZABÓ GÁL ANDRÁS

10. ÁBRA



Hirdetéseinkben a (zárójelben) lévő szám a szerkesztőségi munkát megkönnyítő iktatószám. Tehát nem a cím tartozéka!

C 16, C 116, Plus/4

Commodore 16-os, Plus/4-es felhasználói programokat cserélek. A programlistát a következő címre várom: 2400 Dunaújváros, Alkotás út 10. II/1. (877)

Plus/4 játékprogramokat cserélek! Keresem: CUTHBERT IN THE TOMBS OF DOOM, KICK START, WILD WESTERN, GALAXIONS, ARENA 3000, PETALS OF DOOM, FINGERS MALONE, DEFENCE programokat. Lőrinczi Konrád, 2400 Dunaújváros, Kommunárszk krt. 27. (873)

C 16 játék- és felhasználói programokat cserélek. Wünsch Lajos, 2800 Tatabánya V., Ifjúmunkás u. 17. 3/1. (863)

C 16, Plus/4 játék- és oktatóprogramokat cserélek. Kis István, 3400 Mezőkövesd, Rákóczi u. 39. (865)

C 16 és Plus/4-es játék és felhasználói programot cserélek. Részletes programlistát kérek. Póczik András, 1195 Bp., Nagy S. 24. II/8. (852)

Eladó Plus/4 magnóval, 2 db botkormánnyal, programokkal, felhasználói kézikönyvvel. Jankó Imre, 6726 Szeged, Sas u. 4/1. (874)

C 128 és CP/M alatt futó programok cseréjére társakat keresek. A programok leírásai, kézikönyvei is érdekelnek. Gömöri József, Sümeg TSZ Szakcsoport 1144 Bp., Füredi út 7. (936)

Plus/4 és C 16 angol, orosz nyelvi oktatóprogramok eladók. Kálmán Albert, 3300 Eger, Rákóczi út 31. III/11. Tel. üzenet: 143-031, 330-345 (Bp.) (937)

C 16 és Plus/4-es játék és felhasználói programokat cserélek kazettán. Hegedűs Tamás, 3000 Hatvan, Lehel u. 1. (927)

Commodore 16, 116, Plus/4 programokat cserélek kazettán. Gerencsér András, 7633 Pécs, Építők útja 4/D. (925)

C 16 és Plus/4-es programokat cserélek másikra. Bartucz Géza, 1165 Bp., Mátyásföld, Centenáriumi ltp. Orsika tér 9. fszt. 2. (917)

C 16, C 116, Plus/4 programokat cserélek kazettán. Keresek: Flight Path-737, és assembler fordítót. Mészáros Tamás, 6100 Kiskunfélegyháza, Mártonffy E. u. 2/B. (906)

Új Plus/4-es gépemhez nagy tételben, olcsó áron programokat veszek. Cím: Vida László, 4561 Baktalórántháza, Szőlőskert út 16. (908)

C 16 és Plus/4 programokat cserélek. Karmazin Zoltán, 5001 Szolnok 1. Pf.: 137. (893)

C 128-as programokat cserélnék. Kérem az ajánlott programok nevét és jellegét felsorolni. Maurer Péter, 1182 Bp., Marosvásárhely u. 48. (883)

C 64

C 64 játék- és felhasználói programokat cserélek lemezen és kazettán. Keresek: Scarabeus játék leírását és nyilvántartó prg.-t. (pl. Easy-file) Knyur Géza, 6800 Hódmezővásárhely, Gellért u. 9. (834)

C 64 számítógépemet adatmagnóval együtt bérbe adom. Kovács Gábor, 6800 Hódmezővásárhely, Ady E. u. 21. (840)

C 64-es játékprogramokat cserélnék, listát kérnék. Kb. 800 játék közül lehet választani kazettán. Lukácsi Csaba, 1119 Bp., Nándorfehérvári u. 8/A II. lh. IV/15. (845)

C 64-es játék és egyéb programokat cserélek kazettán. Listát kérek–küldök. Swaczyna Boleslaw, 6800 Hódmezővásárhely, Anna u. 2. (841)

C 64-es felhasználói és játékprogramokat cserélek kizárólag csak lemezen. Sárvári Péter, 4032 Debrecen, Poroszlay 50. (880)

C 64-es felhasználói (zene; grafika; adattároló; szövegszerkesztő) és játékprogramokat cserélnék lemezen. Keresem a Musicalc 1-et. Nagy László, 1122 Bp., Pethényi út 13/A. (878)

C 64-es programokat cserélek kazettán és lemezen. Listát kérek. Nagy Attila, 9330 Kapuvár, Kölcsey u. 8. (926)

C 64-re programokat cserélek. Keresem „Walt Disney Comic Stip Makert" és „The last one" nevű programokat. Cserébe a „Game Makert" és más jó programokat adok. Abonyi Péter, 8400 Ajka, Kohász u. 3. I/3. (902)

Vegyes

„UNIPORT": analóg és digitál villamos jelek fogadása, előállítása. A/D, D/A konverzió C 64-el. Univerzális, megbízható, olcsó! Budapesti Műszaki Egyetem, 664-011/25-86, Bánsághi (816)

C 128-as vagy CP/M üzemmódban adat- és szövegfeldolgozó programokat keresek és cserélek. Ha lehet magyar vagy idegen nyelvű leírással, használati utasítással együtt. Gömöri József, Sümeg TSZ 1144 Bp., Füred út 7. (855)

Kifejlesztettük, gyártjuk C 64 és C 128 gépekhez a vonalkódleolvasó interfészt! Érdeklődőknek felvilágosítást ad az ERFI Általános Elektronika Osztályán Újj Dezső ov. 359-740 (868)

Találmányok, szabadalmak, licencek, szoftverek értékesítésével kapcsolatos tanácsadás, kisvállalkozások szervezése, könyvelése, önrevíziója. Tel.: 167-534. Tagoknak 10% kedvezmény. (942)

Chess Challenger USA Fidelity Electronics 10 fokozatú táblás sakkcomputer eladó. Tel.: 159-850/670 m. (939)

Lehetőleg új Schneider (Amstrad) CPC 664 vagy CPC 6128 (ártól függően) gépet keresek megvételre. Cím: Farkas Attila Jácint, Keszthely, Ságvári köz 5/1 2/4. (934)

Keresem megvételre vagy cserére a GRAFIC BASIC nevű programot lemezemre! Tel.: 328-481. (938)

LD–CF hordozható telepes adatgyűjtő készülék (adatbevitel fényceruzával is) adatkiolvasás közvetlenül számítógéppel * CFI vonalkód olvasó dekóderek fényceruzák illesztéséhez. ERFI, 1027 Bp., Medve u. 25/28. (915)

CBM–8024 típusú printert vásárolnánk. Lehet jó állapotban lévő használt is. Balázs István, Budapesti Harisnyagyár, 1033 Szentendrei út 39–53. Tel.: 889-550/113 (909)

VC 20

Eladó: VC 20 magnóval, játékprogramokkal, 16k-s bővítővel és hozzá való könyvekkel 10 000 Ft-ért. Érdeklődni: 771-352 du. 16 órától. (858)

TÖBB-TÉNYEZŐS DÖNTÉSEK III.

FELÜLMÚLÁSOK

Rövidke sorozatunk befejező részében végre vállalkozunk arra, hogy útmutatást adjunk: hogyan is lehet rangsorolni a döntési lehetőségeket.

Az előző részben már eljutottunk odáig, hogy megállapítottuk, melyik döntési lehetőség múlja felül a másikat. Sőt ennek a felülmúlásnak valamiféle mértékét is megadtuk – azaz beszéltünk arról, hogy az egyik lehetőség gyengén, vagy erősen múlja felül a másikat. Ennek alapján azután felrajzoltunk egy **gráfot** is. Ez egy olyan hálózat, amely pontokból, és az azokat összekötő folytonos, vagy szaggatott nyilakból áll – **ennek segítségével igyekeztünk feltérképezni a döntési lehetőségek egymás közötti kapcsolatát.** Kétféle módon különböztettük meg, hogy egyik döntési lehetőség mennyire múlja felül a másikat: azt mondtuk, hogy ez **a kapcsolat lehet gyenge vagy erős** – és ezt jeleztük azzal, hogy az őket összekötő nyíl (a gráf egy-egy éle) szaggatott vagy folyamatos.
A „gyenge" vagy „erős" felülmúlás, persze, nem számszerűsíthető fogalmak. Nem lehet hozzájuk semmiféle értéket rendelni, aminek alapján eldönthetnénk, hogy végül is melyik lehetőséget válasszuk. Azonban nem is ez a legfőbb gondunk.
Még ha gráfunkban (hálózatunkban) nem is tennénk semmiféle megkülönböztetést a kétféle vonalfajta között, akkor sem tudunk sok okosat kihámozni a rajzból. Legfeljebb valamiféle vizuális képünk lesz a lehetőségekről. Hiszen, ha egy pontból sok nyíl indul ki, akkor az azt jelenti, hogy a ponthoz tartozó döntési lehetőség sok másikat felülmúl – így valószínűleg előnyben kell részesítenünk. Ha pedig sok nyíl fut be egy pontba, az annak a következménye, hogy ezt a döntési lehetőséget sok másik felülmúlja, vagyis vannak nála jobbak is.
Az eddigiek alapján az, aki jó képlátással rendelkezik, már most is jut valamiféle eredményre: ahonnan sok nyíl – ráadásul folytonos! – indul ki, az egy jónak ígérkező döntési lehetőség, ahová pedig sok fut be, az nyilván nem az igazi. Ez azonban nem elég ahhoz, hogy a lehetőségek sorrendjét meghatározzuk.

Vegyük elő újra cipőinket – azaz az előző havi példaprogramunkat, amelyben cipővásárlásról próbáltunk dönteni – és nézzük át az akkor kapott eredményeket! **Az akkori gráfot – vagyis nyilakkal összekötött pontok** együttesét – szöveges formában a következő módon írhatjuk le:
A Fehér cipő gyengén felülmúlja a Feketét.
A Sárgásbarna cipő gyengén felülmúlja a Homokszínűt, a Szürkét és a Fehéret.
A Vörösesbarna cipő gyengén felülmúlja a Homokszínűt, a Szürkét, a Fehéret és a Feketét.
A Sárgásbarna cipő erősen felülmúlja a Homokszínűt, a Szürkét és a Fehéret.
A Vörösesbarna cipő erősen felülmúlja a Szürkét.
Előző számunkban is említettük már, hogy **ha valamelyik lehetőség erősen felülmúlja a másikat, akkor nyilvánvalóan gyengén is felülmúlja azt.** Így talán feleslegesnek tűnik, hogy például azt is megemlítettük: Sárgásbarna gyengén felülmúlja Homokszínűt, Szürkét és Fehéret, holott erősen is megteszi ugyanezt. Ugyanez érvényes akkor is, amikor Vörösesbarna erősen felülmúlja Szürkét – ugyanezt nyilván megteszi gyengén is.
Csupán amiatt „vetemedtünk" ennek külön leírására, mert a felhasznált algoritmus – és így programunk is – először a gyenge felülmúlásokat keresi meg, és később az erőseket. Bizonyára a program felhasználója számára sem érdektelen, hogy lát-e valami képet a közbenső eredményről, vagy sem.

KÖZVETLEN RANGSOR

A rangsorolás legegyszerűbb lehetősége az, hogy **megmérjük, hány elemből áll egy-egy „nyílfolyam", amely kiindult** egy bizonyos döntési lehetőséget jelentő pontból. Azaz milyen hosszú az az út, amit a gráfban innen elindulva bejárhatunk, míg csak zsákutcába nem jutunk.
Hogy ez a nyílfolyam miért jelez bármit is arról, hogy egy döntési lehetőség hol áll a rangsorban? Azért, mert – mint előző számunkban említettük – a gráf **nem feltétlenül tranzitív.** Azaz abból, hogy az A-nak nevezett döntési lehetőség felülmúlja B-t, és B felülmúlja C-t, nem következik az, hogy A felülmúlná C-t is. Így azután a nyíllánc hossza azt mutatja meg, hogy közvetett úton, többszörös áttétellel meddig terjed egy-egy döntési lehetőség hatása.
Először csak az erős felülmúlást jelentő kapcsolatokat vegyük figyelembe –, majd pedig az esetleg itt kialakuló „döntetleneket" eldöntését bízzuk rá a szaggatott vonalú nyilakra – vagyis a gyenge felülmúlásokra.

A példában szereplő gráfunk alapján az eredmény a következőképpen alakul:
1. Sárgásbarna
2. Vörösesbarna
3. Fehér
4. Fekete
4. Homokszín (holtverseny)
4. Szürke

HOSSZÚ ÉS RÖVID UTAK

Első pillantásra úgy tűnhet, hogy a „nyílhosszak mérésének" eljárása meglehetősen egyszerű. Elvégre csak annyit kell tennünk, hogy megkeressük a gráfban a leghosszabb utakat (leghosszabb nyílláncokat), és meghatározzuk ezek méretét. Mostani példánkban – azaz a cipővásárlás kapcsán – tényleg nem ütközünk sok nehézségbe, hiszen csak hatféle lehetőségünk van, és kapcsolataik sem túl „kacifántosak". Mit tehetünk azonban akkor, ha a döntési lehetőségek száma több, és a gráf is kuszább? Hogy tudnánk ezt gépre vinni?

Bemutatjuk most azt a módszert, amelynek segítségével megkereshető a gráf egy pontjából kifutó nyíllánc maximális hossza. Annál is inkább érdemes megismernünk az eljárást, mivel sok más programozási feladatnál is gyakran szükség van hasonló feladat megoldására. Az algoritmus képes arra, hogy a gráf egy pontjából kiinduló nyílfolyam maximális hosszát megadja – mint ezt már említettük –, de kis átalakítással arra is, hogy az egy pontba befutó nyílsorozat hosszának legkisebb értékét határozza meg.

A LEGHOSSZABB UTAK ALGORITMUSA

Az eljárás – egy kis túlzással – talán a mesterséges intelligencia „alapfogásai" közé sorolható.
Már az előző számunkban közölt programrészletben definiáltunk egy EREDM nevű tömböt. Ennek elemei 1, vagy 0 értéket vehetnek fel. Ha az I-edik lehetőség felülmúlja a J-ediket, akkor EREDM (I, J) értéke 1 lesz, ha nem, akkor 0. (Egyelőre nem teszünk megkülönböztetést aközött, hogy a döntési lehetőségek erősen, vagy gyengén múlják felül egymást.)
A mátrix (tömb) mérete nyilvánvalóan NDO*NDO, azaz sorainak és oszlopainak száma is a döntési lehetőségek számával egyenlő. Ha mátrixunk készen van, akkor sorról sorra, ezen belül pedig oszlopról oszlopra – azaz programban: egy kettős ciklussal – végignézzük az elemeit. Amikor 1-est találunk, vagyis EREDM (I, J) = 1, akkor megnézzük, hogy a J-edik lehetőség felülmúl-e egy másikat. Ha nem, akor J most a végállomás, így továbbléphetünk. Ha igen, akkor mindaddig keresünk, míg az I-ből kiinduló út végállomását meg nem találjuk. Ezt azután J1-gyel jelöljük. Most már csak annak megállapítása van hátra, hogy vajon ez-e a leghosszabb út I-ből J1-be? Ennek érdekében vizsgálatunkat újra

```
10 REM ********************************
20 REM *     C= UJSAG SORSZAM 076     *
30 REM *     TOBBTENYEZOS DONTESEK    *
40 REM *     PROGRAM: TALLER JOZSEF   *
50 REM ********************************
100 REM ERTEKELESI TENYEZOK SULYOZASA
110 POKE53281,0:PRINT"■":PRINT
115 PRINT"■          **** ADATBEVITEL ****":PRINT
120 INPUT"   ■ ERTEKELESI TENYEZOK SZAMA ";NERT: PRINT"■":PRINT
125 IF NERT>20 THEN 120
130 DIMERT$(NERT): DIM SULY(NERT):DIMA(NERT):DIM AN(NERT):DIM PREF(NERT,NERT)
140 FORI=1TONERT
150 PRINTI;"■. TENYEZO ";:INPUT ERT$(I)
155 IFLEN(ERT$(I))>12THEN ERT$(I)=LEFT$(ERT$(I),12)
160 NEXTI:PRINT
165 PRINT"■":PRINT"■           **** ADATBEVITEL ****":PRINT
170 F=NERT*(NERT-1)/2
180 FORI=1TOF:A$=A$+CHR$(I):NEXTI
190 FORI=FTO1STEP-1
200 X=INT(I*RND(0))+1
210 A$=LEFT$(A$,X-1)+RIGHT$(A$,F-X)+MID$(A$,X,1)
220 NEXTI
225 PRINT"     ■ MELYIK FONTOSABB ?   (A/B)■":PRINT
230 FORQ=1TOF
240 P=ASC(MID$(A$,Q,1))
250 I=INT(1.5+SQR(2*P))
260 J=P-(I-1)*(I-2)/2
270 X=RND(0)
280 IF X<.5THEN R=I:I=J:J=R
290 PRINT ERT$(I);TAB(13);"(A)    ";ERT$(J);TAB(32);"(B)";
310 INPUT Q$:PRINT
320 IFQ$="A"THEN PREF(I,J)=1:A(I)=A(I)+1:GOTO 350
330 IFQ$="B"THEN PREF(J,I)=1:A(J)=A(J)+1:GOTO 350
340 GOTO290
350 NEXTQ
360 FORI=1TONERT:AN=AN+A(I)*A(I):SULY(I)=(A(I)+.5)/NERT:NEXTI
370 D=NERT*(NERT-1)*(2*NERT-1)/12-AN/2
375 PRINT"■":PRINT"■           **** ERTEKELES ****":PRINT
380 PRINT:PRINT"■       KORHARMASOK SZAMA :■";D
390 IF2*INT(NERT/2)=N THENK=1-24*D/(NERT*(NERT*NERT-4)):GOTO410
400 K=1-24*D/(NERT*(NERT*NERT-1))
410 PRINT:PRINT"■        KOVETKEZETESSEG :■";INT(K*100);"%"
415 IFD=0THEN WAIT198,255:GETA$: GOTO440
420 PRINT:PRINT:INPUT"■        KERI A KORHARMASOKAT ■(I/N)";Q$
430 IFQ$="I" THEN GOSUB500
440 PRINT"■":PRINT:PRINT"■ *** AZ ERTEKELESI TENYEZOK SULYAI ***":PRINT:PRINT
450 FORI=1TONERT
460 PRINT"■   ";ERT$(I);TAB(25);"■";INT(100*SULY(I));"%"
470 NEXTI
480 GOTO1000
500 REM KORHARMASOK
510 PRINT"■":PRINT:PRINT"■           **** KORHARMASOK ****■":PRINT
530 FORM=1TONERT:K=NERT+1:A=NERT
540 FORI=1TONERT
550 IF A(I)<ATHENA=A(I):K=I
560 NEXTI
570 FORJ=1TONERT
580 IFPREF(K,J)=1THEN GOSUB650:PREF((K,J)=0
590 NEXTJ:A(K)=NERT
600 FORS=1TONERT:PREF(S,K)=0:NEXTS
610 NEXT M:WAIT198,255:RETURN
650 FORL=1TONERT
660 IFPREF(J,L)=0 OR PREF(L,K)=0THEN670
665 PRINTTAB(15);ERT$(L):PRINTTAB(15);ERT$(K):PRINTTAB(15);ERT$(J):PRINT
670 NEXTL:RETURN
1000 REM EGYMAST FELULMULO DONTESEK
1005 WAIT 198,255:GETA$
1010 PRINT "■■         **** ADATBEVITEL ****":PRINT
1020 INPUT"■    DONTESI LEHETOSEGEK SZAMA ";NDO:PRINT:PRINT
1025 IF NDO>20THEN1020
1030 DIM DO$(NDO)
1040 FORI=1TONDO
1050 PRINTI;"■. LEHETOSEG ";:INPUT DO$(I)
1055 IF LEN (DO$(I))>12THENDO$(I)=LEFT$(DO$(I),12)
1060 NEXTI:PRINT
1065 PRINT"■":PRINT"■           **** ADATBEVITEL ****":PRINT
1070 SIGMA=0:FORI=1TONERT:SIGMA=SIGMA+SULY(I):NEXTI
1080 DIMDO(NDO,NERT):DIMNO(NDO,2,2):DIMRANG(NDO,3)
1090 FORI=1TONDO
1100 PRINT:PRINT"■";DO$(I):PRINT
1110 FORJ=1TONERT
1120 PRINT"■ OSZTALYZATA ";ERT$(J);" SZERINT ";:INPUTDO(I,J)
1130 NEXTJ,I
1140 PRINT"■":PRINT:PRINT"■        **** P MATRIXOT SZAMITOM ****":PRINT
1150 DIM P(NDO,NDO):DIMDMAX(NDO,NDO):DIMEREDM(NDO,NDO,2):DIM E(NDO,NDO,2)
1160 FORI=1TONDO:FORJ=1TONDO
1170 P(I,J)=0:P(J,I)=0
1180 FORK=1TONERT
1190 DI=DO(I,K):DJ=DO(J,K)
1200 IF DI>DJTHENP(I,J)=P(I,J)+SULY(K)
1210 IF DI<DJTHENP(J,I)=P(J,I)+SULY(K)
1220 NEXTK
1230 IF P(J,I)=0THENP(J,I)=.0001
1235 IF P(I,J)=0THENP(I,J)=.0001
1240 NEXTJ,I
1250 BC=0
1260 DIM D1(NERT):DIM D2(NERT)
1270 PRINT"■":INPUT"C1■ ";C1:PRINT:IFC1<0ORC1>1THEN1270
1272 PRINT"■":INPUT"C2■ ";C2:PRINT:IFC2<0ORC2>C1THEN1272
1274 PRINT"■":INPUT"C3■ ";C3:PRINT:IFC3<0ORC3>C2 THEN1274
1280 IF BC=0THEN1310
1290 PRINT:INPUT"AKARJA MODOSITANI A D1 ES A D2 KUSZOBO-               KET (I/N)";RE
P$
1300 IF REP$="N"THEN1360
1310 PRINT:PRINT:FORI=1TONERT
1320 PRINT"■";ERT$(I);" TENYEZO SZERINT"
1330 INPUT"■D1 ";D1(I)
1340 INPUT"■D2 ";D2(I):IFD2(I)>D1(I)THEN1340
1350 NEXTI
1360 PRINT"■":PRINT"■  **** FELULMULASOK MEGHATAROZASA ****"
1370 K1=1:K2=2:BC=1
1380 FORI=0TONDO
1390 FORJ=0TONDO
1400 FORK=K1TOK2
1410 EREDM(I,J,K)=0
1420 NEXTK,J,I
1430 FORI=K1TONDO-K1
1440 FORJ=I+K1TONDO
1450 DMAX(I,J)=0:DMAX(J,I)=0
1460 FORK=K1TONERT
1470 DK=DO(J,K)-DO(I,K)
1480 IFDMAX(I,J)=K2THEN1510
1490 IFDK>D2(K)THENDMAX(I,J)=K2:GOTO1510
1500 IFDK>D1(K)THENDMAX(I,J)=K1
1510 NEXTK
1520 RP=P(I,J)/P(J,I):RM=K1-P(I,J)/SIGMA:IF RP=0ANDP(J,I)=.0001THEN1570
1530 IFRP<K1THEN1690
1540 IFRM<C1THEN1600
1550 IFDMAX(I,J)=K2THEN1630
1560 EREDM(I,0,K1)=K1:EREDM(I,0,K2)=K1:EREDM(I,J,K1)=K1
1565 EREDM(I,J,K2)=K1:EREDM(0,J,K1)=K1:EREDM(0,J,K2)=K1
1570 NEXTJ
1580 NEXTI
1590 GOTO 1810
1600 IFRM<C2THEN1650
1610 IFDMAX(I,J)=0THEN1560
1620 GOTO1660
1630 IF RP=K1THEN1690
1640 GOTO1570
1650 IFRM<C3THEN1630
1660 IFDMAX(I,J)=K2THEN1630
1670 EREDM(I,0,K2)=K1:EREDM(0,J,K2)=K1:EREDM(I,J,K2)=K1
1680 GOTO1570
1690 RM=K1-P(I,J)/SIGMA
1700 IF RM<C1 THEN1740
1710 IFDMAX(J,I)=K2THEN1570
1720 EREDM(J,0,K1)=K1:EREDM(J,I,K1)=K1:EREDM(0,I,K1)=K1:EREDM(J,0,K2)=K1
1725 EREDM(J,I,K2)=K1:EREDM(0,I,K2)=K1
1730 GOTO1570
1740 IFRM<C2THEN1770
1750 IFDMAX(J,I)=0THEN1720
1760 GOTO1570
1770 IFRM<C3THEN1570
1780 IFDMAX(J,I)=K2THEN1570
1790 EREDM(J,0,K2)=K1:EREDM(J,I,K2)=K1:EREDM(0,I,K2)=K1
1800 GOTO1570
1810 FORK=K2TOK1STEP-K1
1820 FORI=0TO NDO:IJ=EREDM(I,0,K)
1825 IFK=K1ANDI<>0ANDIJ<>0THENPRINT"■";DO$(I);" EROSEN FELULMULJA:"
1827 IFK=K2ANDI<>0ANDIJ<>0THENPRINT"■";DO$(I);" GYENGEN FELULMULJA:"
1830 FORJ=0TO NDO
1840 E (I,J,K)=EREDM(I,J,K)
1850 IF EREDM(I,J,K)=1 ANDI*J<>0   THEN PRINT DO$(J)
1860 NEXTJ
1870 WAIT 198,255:GETA$:PRINT:NEXTI:NEXTK
```

```
2000 PRINT"■":PRINT   "■          **** LEGHOSSZABB        ****"
2005 AZ=1:BZ=0:GOSUB3000
2010 FORI=1TO NDO
2020 X=NO(I,K1,K1):A$=DO$(X):GM=INT(NO(I,K2,K1))
2030 PRINT:PRINT "■ HOSSZ";A$;"-TOL:■";GM;" ES ";NO(I,K2,K2)
2040 NEXTI
2050 RANG(K1,K1)=NO(K1,K1,K1)
2060 RANG(K1,K2)=K1
2070 FORI=K2TONDO
2080 IF NO(I,K2,K1)<NO(I-K1,K2,K1)THEN2110
2090 RANG(I,K2)=RANG(I-K1,K2)
2100 GOTO2120
2110 RANG(I,K2)=I
2120 RANG(I,K1)=NO(I,K1,K1)
2130 NEXTI
2132 FORK=K1TOK2
2134 FORI=0TONDO
2136 FORJ=0TONDO
2140 EREDM(I,J,K)=E (J,I,K)
2150 NEXTJ:NEXTI:NEXTK
2160 AZ=-1:BZ=NDO:GOSUB3000
2170 NO(0,K2,K1)=.00001
2180 FORI=K1 TO NDO
2190 IF NO(I,K2,K1)<NO(I-K1,K2,K1)THEN2210
2200 NO(I,K2,K2)=NO(I-K1,K2,K2):GOTO2220
2210 NO(I,K2,K2)=I
2220 FORJ=K1 TO NDO
2230 IF RANG(J,K1)=NO(I,K1,K1)THEN2250
2240 NEXTJ
2250 RANG(J,K2)=RANG(J,K2)+NO(I,K2,K2)
2260 NEXTI
2270 FORI=K1TONDO-1
2280 FORJ=I+1TONDO
2290 IFRANG(I,K2)<=RANG(J,K2)THEN2320
2300 Z=RANG(I,K1):RANG(I,K1)=RANG(J,K1):RANG(J,K1)=Z
2310 Z=RANG(I,K2):RANG(I,K2)=RANG(J,K2):RANG(J,K2)=Z
2320 NEXTJ:NEXTI
2340 RANG(K1,3)=K1
2350 FORI=K2TONDO
2360 IFRANG(I,K2)>RANG(I-1,K2)THEN2390
2370 RANG(I,3)=-ABS(RANG(I-K1,3))
2380 GOTO2400
2390 RANG(I,3)=I
2400 NEXTI
2410 PRINT"■":PRINT"■          **** A DONTESEK RANGSORA ****"
2420 FORI=K1TONDO
2430 PRINT"■":  PRINTABS(RANG(I,3));". ";DO$(RANG(I,K1));
2440 IFRANG(I,3)<0THENPRINT"    (HOLTVERSENY)";
2450 NEXTI
2460 PRINT:PRINT"■":INPUT"MEGISMETLI";R$
2470 IFR$="I"THEN1270
2480 END
3000 REM A LEGHOSSZABB UTAK KERESESE
3010 FORK=K1TOK2
3020 B=0
3030 FORI=K1TONDO
3040 FORJ=K1TONDO
3050 RR=EREDM(I,J,K)
3060 IFRR<=0THEN3210
3070 B=K1
3080 IFEREDM(J,0,K)<>0THEN3110
3090 EREDM(I,J,K)=-RR
3100 GOTO3210
3110 I1=J
3120 B1=0
3130 FORJ1=K1TONDO
3140 RT=EREDM(I1,J1,K)
3150 IFRT=0THEN3190
3160 IFRT>0THENB1=K1:GOTO3190
3170 VA=RR-RT
3180 IFVA>ABS(EREDM(I,J1,K))THENEREDM(I,J1,K)=-VA
3190 NEXTJ1
3200 IFB1=0THENEREDM(I,J,K)=0
3210 NEXTJ
3220 NEXTI
3230 IFB=K1 THEN3020
3240 NEXTK
3250 FORK=K1TOK2
3260 FORI=K1TONDO
3270 MM=0
3280 FORJ=K1TONDO
3290 IFABS(EREDM(I,J,K))>MMTHENMM=ABS(EREDM(I,J,K))
3300 NEXTJ
3310 NO(I,K1,K)=I:NO(I,K2,K)=MM
3320 NEXTI
3330 NEXTK
3340 FORI=K1TONDO
3350 NO(I,K2,K1)=(NO(I,K2,K1)+NO(I,K2,K2)/1000)*AZ+BZ
3360 NEXTI
3370 FORI=K1TONDO-K1
3380 FORJ=I+K1TONDO
3390 IFNO(I,K2,K1)>NO(J,K2,K1)THEN3440
3400 FORK=K1TOK2
3410 Z=NO(I,K1,K):NO(I,K1,K)=NO(J,K1,K):NO(J,K1,K)=Z
3420 Z=NO(I,K2,K):NO(I,K2,K)=NO(J,K2,K):NO(J,K2,K)=Z
3430 NEXTK
3440 NEXTJ
3450 NEXTI
3460 RETURN
```

kezdjük I és J1 között. Eközben mindig megjegyezzük a két pont között mért leghosszabb nyílfolyam méretét, és beírjuk az EREDM (I, J) megfelelő mezőjébe.
Amikor már minden lehetőséget végigvizsgáltunk, vagyis nagyobb hosszat nem találhatunk, akkor az EREDM mátrix megfelelő értékét negatívra váltjuk – ezzel jelezve, hogy itt befejeztük a vizsgálódást. Ha azután az EREDM mátrix minden eleme negatív vagy 0 lesz, akkor véget ér eljárásunk: a mátrix minden eleme megmutatja – persze, ellenkező előjellel –, hogy két pont között mekkora a leghosszabb út. **Ezt a módszert illusztrálja blokksémánk.** Az érdeklődők már most is vethetnek az ábrára egy-két pillantást, hogy jobban megértsék az őket érdeklő részleteket – de ennek igazán akkor lesz csak értelme, ha megismerkedünk a második rangsorolási módszerrel is.

KÖZVETETT RANGSOR

Más lehetőségünk is van a döntések rangsorolására, mint a korábban bemutatott közvetlen módszer. **Most ne a kiinduló nyílfolyamot vizsgáljuk meg, hanem azt a nyílsorozatot, ami egy adott döntést jelképező pontba beérkezik.** Vagyis nézzük meg azt, hogy egy-egy döntési lehetőséget a többi mennyire, és mennyiben múl felül – és eszerint is osztályozzuk őket. Úgy is fogalmazhatnánk: most az egyes döntési lehetőségek „alulmúlását” értékeljük. Itt persze ismét csak azt használjuk ki, hogy gráfunk nem tranzitív – akárcsak az imént. Egy döntési lehetőségről ugyanis az is sokat mondhat, hogy az őt jelképező pontba milyen hosszú nyíllánc fut be. Ez arról ad információt, hogy a vizsgált lehetőséget más döntések hányszoros áttételen keresztül múlnak felül. Ennél a fordított módszernél persze az a döntési lehetőség kerül előbbre a rangsorban, amelybe rövidebb nyílfolyam érkezik. Újra csak hasonló módszert követünk, mint korábban – azaz ismét a nyílfolyamok hosszát tekintjük mérvadónak, és először most is a folytonos nyilakat vizsgáljuk, majd ezután a szaggatottakat.
Most már tényleg érdemes áttanulmányozni a blokkdiagrammot is, mert programunkban egy eszerint működő szubrutin fogja meghatározni mind a kiinduló, mind a beérkező leghosszabb utak hosszát. Így már érthetővé válik a B változó szerepe is, vagyis az, hogy miért csináljuk végig kétszer szinte ugyanazt. Azért, mert először az egyes döntési tényezőkből kiinduló leghosszabb nyílláncot keressük, és aszerint rangsorolunk, másodszor pedig a befutó nyílfolyamok hosszát, és annak alapján állítunk fel sorrendet.

ÖSSZESÍTETT RANGSOR

Jogosan kérdezi az olvasó, hogy mi sül ki végül is ebből? Elvégre nekünk egyetlen rangsorra van szükségünk – ezt is ígértük sorozatunkban idáig –, most pedig kapásból két lehetőséget is felsoroltunk, amelyek képesek valamiféle értékrend előállítására. A következőkben bemutatjuk, hogy

a kétféle rangsort lehet – sőt kell is – ötvözni. Az algoritmus végrehajtása, illetve a program futtatása során ugyanis a legutolsó dolgunk az, hogy az említett kétféle rangsor **átlagát** vesszük, és ebből alakul ki végül is a döntési lehetőségek érvényes sorrendje.

Így azután a példában szereplő cipők sorrendje a következő módon alakul:

1. Vörösesbarna } (holtverseny)
1. Sárgásbarna
3. Fehér } (holtverseny)
3. Fekete
5. Szürke } (holtverseny)
5. Homokszín

Látható tehát, hogy ha igazán jó, minden szempontból megfelelő cipőt akarunk venni, akkor vagy a Vörösesbarnát, vagy a Sárgásbarnát kell megvásárolnunk. Hogy végül is a kettő közül melyik mellett döntsünk?

Két lehetőségünk van. Ha már nagyon elkopott az eddig hordott cipőnk, és nincs türelmünk tovább elemezni a döntési lehetőségeket, akkor a hasunkra csapunk, és a két első helyezett közül azt vesszük meg, ami ezután a kezünk ügyébe esik. Ha azonban még mindig szeretnénk valamelyest objektív módon dönteni, akkor kihasználjuk azt, hogy programunknak van egy ismétlési funkciója. Itt az egyes döntési tényezőket már nem kell újra beadnunk, de a c és d értékeket tetszés szerint megváltoztathatjuk. (E változók szerepét, illetve módosításuk hatását előző számunkban részletesen ismertettük.)

Példánkból kiderül, hogy **módszerünk sem ad mindig egyértelmű rangsort,** hiszen most is három holtversenyt találunk a hat lehetőség között. Így általánosítható is a fenti tanács: vagy hasraütéssel választunk (amit persze már a kezdet kezdetén is megtehettünk volna...), vagy pedig a c és d paraméterek szűkebbre, vagy tágabbra állításával újra próbálkozunk.

A PROGRAMRÓL

Mivel már a leírás során bemutattuk a blokksémát, a program működéséről csak kevés mondanivalónk lehet.

Amit mindenekelőtt meg kell jegyeznünk: mostani programlistánk tartalmazza a sorozat előző két részében közölt programrészleteket is, így ez a lista teljes. Aki már bepötyögte az eddigieket, az most csak a 2000-es sortól fűzze hozzá a folytatást.

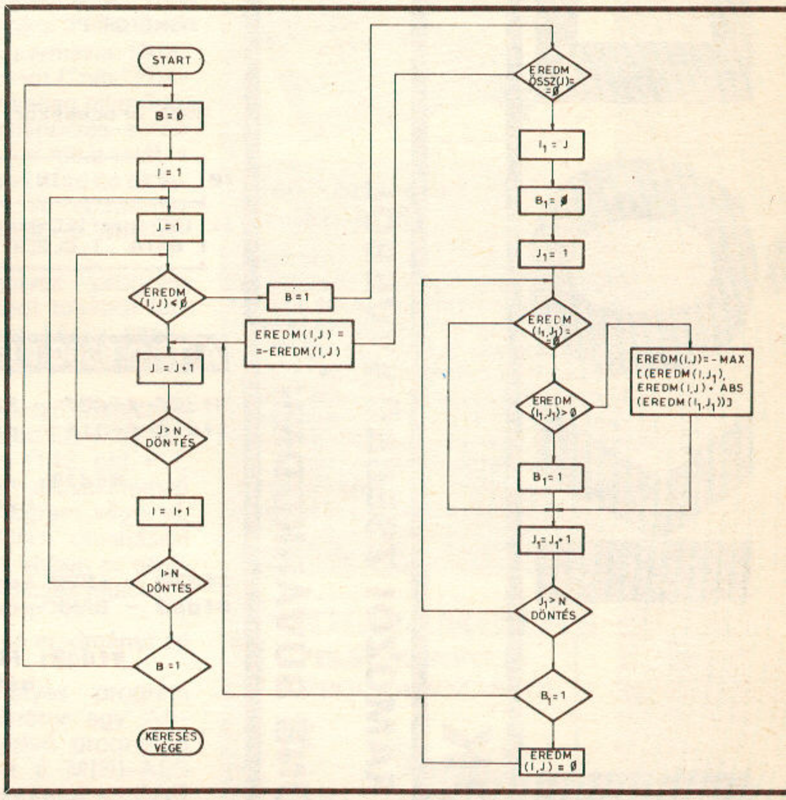
A programban a leghosszabb utak meghatározását a 3000-es szubrutin végzi. (Ennek működésére nem is térünk ki külön, mert a blokkdiagram illusztrálja). A kétféle rangsor miatt kétszer is meghívjuk a szubrutint: először rögtön a 2005-ös sorból, másodszor pedig a 2160-asból.

A 2010–2040 sorokban ki is írjuk, hogy az egyes döntési lehetőségektől milyen hosszú út vezet valamely más lehetőségig. Mindezt azért, hogy már idejében lássuk, ha rosszul döntöttünk a c és a d értékek megválasztásánál – hiszen ilyenkor túl sok az azonos hosszúságú út.

2050-től 2150-ig alakítjuk ki a közvetlen rangsort.

A 2160-as sorból – mint már írtuk –, újra csak meghívjuk a 3000-es szubrutint, amely a leghosszabb utakat keresi. Azután 2400-ig egyesítjük e két rangsort, majd 2410 és 2450 között kinyomtatjuk. Végül a 2460–2480 sorok szolgálnak arra, hogy ha ismételni akarunk, akkor ezt megtehessük.

Tallér József



TÖBB-TÉNYEZŐS DÖNTÉSEK III.

FELÜLMÚLÁSOK

GÉPI KÓD

KERNAL TÁBLÁZATOK

VADNAI SZABOLCS C 16-OS PROGRAMOZÓI ZSEBKÖNYVÉBŐL – AMELY VÉGRE MEGJELENT – EZÚTTAL A KERNAL KÖRÜL KEZDÜNK BÚVÁRKODNI!

tartalom	név	hex	dec
aktív logikai file# tábla	LAT	$509-$512	1289-1298
készülékszám-tábla	FAT	$513-$51c	1299-1308
másodlagos-cím (csatorna#) tábla	SAT	$51d-$526	1309-1318
OS memória-kezdőcím ptr	MEMSTR	$531,$532	1329,1330
OS memória felső-határ	MEMSIZ	$533,$534	1331,1332
hardware IRQ interrupt vektor	CINV	$314,$315	788,789
BRK utasítás interrupt vektor	CBINV	$316,$317	790,791
nem maszkolható interrupt vektor	NMINV	$312,$313	786,787
ST I/O státusz-szó	STATUS	$090	144
real-time jiffy-clock	TIME	$a3- $a5	163-165
MONITOR ST, A, X, Y regiszterek		$554-$557	1364-1367
MONITOR SP-reg		$558	1368
MONITOR PC		$552,$553	1362,1363

8501 processzor I/O port-ja $0000-$0001 0-1

$0	IN	IN	OUT	IN	OUT	OUT	OUT	OUT
$1	IEC-bus DATA	IEC-bus CLOCK		tape sense	tape motor	IEC-bus ATN	IEC-bus CLOCK	IEC-bus DATA

A TED 8360 REGISZTEREI:

```
$fd00-$fd07 - RS232 I/O
$fd10-$fd1f - paralell-port

        $fd10: szalagegység
              bit 2 - 0 : PLAY ON
                      1 :      OFF

$fd30 - 6529 keyboard-port (oszlop kijelölése)
$fdd0 - BANK-port

        $fdd0: ROM-enable
              bit 0-1  -  00 : BASIC
                          01 : PLUS-3
                          10 : C1 low
                          11 : C2 low
                  2-3  -  00 : KERNAL
                          01 : PLUS-3
                          10 : C1 high
                          11 : C2 high

$fec0-$fec5,$fef0-$fef5 - paralel IEC I/O
$ff00-$ff1f - hang/video/timer/interrupt/keyboard regiszterek
              (lsd. a ZENE és a GRAFIKA c. fejezeteket)
$ff3e-$ff3f - ROM enable/disable
```

KERNAL STÁTUSZOK:

Ha a KERNAL-rutinok végrehajtása során hiba következik be, a carry-flag ON-ba kerül és az A regiszter a hibakódot fogja tartalmazni. Lehetséges hibakódok:

0 – a rutin félbeszakítását a STOP gombbal kezdeményezték
1 – túl sok file van megnyitva
2 – a file már nyitva van
3 – a file nincs megnyitva
4 – a file-t nem találja
5 – az egység nincs jelen
6 – a file nem input-file
7 – a file nem output-file
8 – a file-név hiányzik
9 – illegális készülékszám

A KERNAL rutinokat összefoglaló táblázatot októberi számunkban közöljük.



ADALÉK

A Data Becker–Novotrade Rt. Compiler c. könyvével kapcsolatban több Olvasónak problémát okozott a 143. old. P.4. programja. Látszólag valóban hiányzik a szemantika, elemző és kódgeneráló program utolsó része (161–). Ez a német – eredeti – kiadványban is így szerepel.
Annak az olvasónak, aki szisztematikusan és figyelmesen végigolvasta a könyvet, világossá vált, hogy ez a hiányzó szubrutin egy fejlettebb formában már a 4.8 alfejezetben ismertetésre került.

MIRŐL IS VAN SZÓ?

A hiányzó szubrutin a MINIATUR–SYN tartalmát hivatott kinyomtatni, hogy a kapott kódsorozatot a szemantikai szimbólumok táblázatának (4.6 alfejezet, 97. old.) segítségével visszafejthessük.
Most vizsgáljuk meg a 4.8 alfejezetben (118. old. P.2. program) tárgyalt programot. A program a MINIATUR–SYN tartalmát a szemantikai szimbólumok táblázatának segítségével dekódolva, olvasható, (szöveges) formában elemzi az általunk MINIATUR nyelven megírt programot, megkímélve bennünket a manuális dekódolás fáradságos munkájától.
A kapott lista alapján mi is könnyen ellenőrizhetjük az általunk írt programot, szemantikailag helyes-e (pl. definiáltunk-e minden azonosítót; nincs-e valamelyik azonosító, változó újra definiálva stb.)
Hogyan használjuk a könyvben ismertetett compiler programrendszert? Első lépésben a gép szövegszerkesztőjével megírjuk a kívánt MINIATUR programot, vagy felhasználjuk a könyv valamelyik mintapéldáját, amit SAVE "név",8 paranccsal lemezre írunk.
A compiler első menetében a PARSER (4.7. alfejezet 102. old. P.1. program) szintaktikai ellenőrző program a megírt, és lemezen tárolt MINIATUR programot ("név") beolvassa és szintaktikailag ellenőrzi. Amennyiben az ellenőrzés során bármilyen szintaktikai hibát talál, a hibára utaló üzenetet küld és befejezi az ellenőrzést. A hibaüzenet alapján ki kell javítani a MINIATUR programot.
Szintaktikailag helyes program esetén a programból további feldolgozásra alkalmas kódot generál, és a MINIATUR–SYN file-ba írja.
Az ellenőrzés következő szakasza a szemantikai vizsgálat. Ezt a feladatot az 5. fejezetben (143. old. P.4. pr.) ismertetett szemantikai elemző és kódgeneráló program végzi el. A programot <RUN> utasítással indítsuk, mert csak ebben az esetben végzi el a szükséges feladatokat.
Hiba esetén hasonló eredményt ad a PARSER-hez.
Szemantikailag helyes program esetén a kódgenerátor egy ASSEMBLY forrásnyelvű programot hoz létre, amelyet a MINI–ASS file-ba ír (ezt a programot <LOAD "MINI–ASS",8> paranccsal be lehet tölteni a gépbe tanulmányozás céljából).
Az így létrehozott ASSEMBLY programot bármilyen ASSEMBLER (6502, 6510) segítségével futtatható programmá fordíthatjuk. A könyvben bemutatott ASSEMBLER a létrehozott programot ellátja egy COPYRIGHT felirattal is, de ettől eltekintve bármilyen ASSEMBLY fordítására alkalmas.
Az ASSEMBLER szintén ellenőrzi a fordítandó programot – a kódgenerátor által létrehozott program helyes – és hasonlóan jár el az előző programokhoz. Az ASSEMBLER a létrehozott programot a felhasználó által kijelölt "név" nevű file-ba tölti.
(Az ellenőrzés és a kódgenerálás az összes program esetében párhuzamosan történik.)
A compiler-rendszer hibamentes befejeződése után a kész programot <LOAD "név",8,1>:<RUN> parancsokkal futtathatjuk.

Bartha Imre



Első könyvem a mikrókról

(Usborn)
A mikrók, vagyis a számítógépek egyre több magyar családban jelentenek ma már tanulást és játékot. Ez egy igazi képeskönyv a gyerekeknek, amelyből mesélhetünk is nekik. Sok szülő maga is gondban van, ha gyermek számítástechnikai kérdéseket tesz fel. Ez a könyv segít abban is, hogy gyermekeink okos kérdéseire okos válaszokat is adhassunk.

Ára: 99,– Ft



Smith:

Első könyvem a programozásról

(Usborn)
Legendák terjedtek el arról, hogy a gyerekek milyen ügyesek a programozásban. Mint minden legenda, ez is csak félig igaz, a gyerekeknek szükségük van támaszokra, hogy valóban megértsék, mit is csinálnak. Ebben a könyvben a szinte minden gépen futtatható programot igazán gyerekekhez szóló magyarázatok és rengeteg rajz, ábra és grafika egészíti ki. Minden alapvető tudnivaló sorra kerül: hogyan működik a számítógép, hogyan kell utasításokat adni a gépnek, mi történik ezek hatására.

Ára: 99,– Ft

KUKKOLÓ

MI AZ A MONITOR?

CSAK KEZDŐKNEK

A számítástechnika mélységeiben járatlan olvasó rovatunkból már megismerkedhetett a bináris és a hexadecimális számokkal, valamint megtudhatta azt is, hogy ezeket hogyan is tárolja a számítógép. A korábbiakban arra is fény derült, hogy a memória egyes helyein tárolt számoknak más és más szerepük lehet attól függően, hogy ez a szám pontosan hol helyezkedik el a memóriában, illetve, hogy milyen más tárolt értékekkel van kapcsolatban.
Eddigi programjainkat is C-16-ra, illetve C Plus/4-re írtuk, és így akartunk segítséget nyújtani ahhoz, hogy kiki ráérezzen: mi is az a monitorprogram. „Ráérezzen" írtuk, elvégre az eddigiek nem igazán voltak azok, csak érzékeltetni kívánták a lényeget.

Akinek Commodore 16-os, vagy Plus/4-es gépe van, az nyilván tudja, hogy gépe rendelkezik monitor-üzemmóddal is. Mostanáig talán csak kellő bátorsága nem volt a kezdő felhasználónak ahhoz, hogy ezt ki is próbálja.
Mit lehet tulajdonképpen egy monitorprogrammal csinálni? Meg lehet vizsgálni bármelyik memóriacím tartalmát; össze lehet hasonlítani két memóriaterületet és kiíratni a különbségeket; gépi kódú programot emberközeli, assembly nyelven listázhatunk vele; adott értékeket kereshetünk ki segítségével a memóriában és – nem utolsósorban – megvizsgálhatjuk a processzor regisztereinek a tartalmát, amelyek rendkívül fontosak a gép működése szempontjából. Most ezeknek a regisztereknek a működésével ismerkedünk meg.

MI A REGISZTER?

Regiszternek nevezzük azokat a tárhelyeket, amelyeket a gépet vezérlő fő alkotórésze, a mikroprocesszor működése során használ. A gépi kódú utasítások végrehajtása során – egy kivétellel (NOP) – valamelyik regiszter tartalma megváltozik, vagy értékét átadja valamelyik memóriacímnek.

Hogy a dolog könnyebben érthető legyen, helyettesítsük számítógépünket egy „sokfiókos" szekrénynyel, amely 65535 számozott fiókkal rendelkezik. Egy-egy fiók egyegy memóriacímnek felel meg. (Egy 64 K-s számítógép ennyi memóriacímet kezel.) Nevezzünk ki egy „hivatalnokot", akinek az a feladata, hogy egy számára megadott előírás alapján kezelje a szekrényt, vagyis ha kell, rakjon bele valamit a kijelölt fiókba, ha kell, akkor pedig vegyen ki belőle valamit. (A „hivatalnok" az maga a mikroprocesszor, az előírás pedig a program.)
Hogy megkönnyítsük a „hivatalnok" munkáját, adjunk neki egy íróasztalt, amelyen hat doboz van, amelyekbe áttekinthetően, hozzáférhetően el tudja helyezni a rá bízott dolgokat. Nem nehéz kitalálni, hogy ezek a dobozok maguk a regiszterek.

BELÉPÉS

Akkor most nézzük mit láthatunk a regiszterek tartalmából a monitorban. Ha a gépnek kiadjuk a MONITOR parancsot, a következő üzenet jelenik meg:

```
  PC    SR   AC   XR   YR   SP
 0000   00   00   00   00   F8
```

Ugyanezeket a sorokat láthatjuk akkor is, ha már MONITOR üzemmódban vagyunk és R parancsot adunk, vagy ha egy gépi kódú program futás közben BRK utasítást talál, ami azt jelenti, hogy a program futása nem folytatódik. Ezt igen gyakran tapasztalhatják azok, akik gondatlanul kalózmásolt játékprogramot próbálnak futtatni. **Fontos tudnunk, hogy a gépi kódú BRK és az általunk jól ismert RUNSTOP nem ugyanaz. Maradva a hivatalnok és az előírás példájánál, a RUNSTOP egy RUNSTOP feliratú előírás megérkeztét jelenti emberünknek, aki unatkozva végre is hajtja a megfelelő cselekvéssorozatot. A BRK ezzel szemben egy, az előírásból hiányzó lap! A sorszáma megvan, de nincs rajta semmi. Mondhatnánk, hogy vegye a következő lapot emberünk, de erre sajnos nem képes, hiszen az üres lapon az az utasítás sincs rajta legalább, hogy vegye elő a következő lapot! (Szegény hivatalnokunknak ilyesmi magától eszébe sem juthat!)**

Visszatérve a monitorban látottakra:
A felső sor betűi a regiszterek nevei. Az alsó sorban a regiszterek tartalmát láthatjuk. Egy-egy számot, hexadecimális formában. (Most talán elkél egy kis ismétlés?!) Meg kell jegyezni, hogy ezek csak a BRK után és az R monitor parancs után jelzik ki a regiszterek aktuális értékét, egyébként normál esetben a MONITOR utasítás után az utolsó monitorozás végén eltárolt értékeket láthatjuk. Nézzük meg egyenként, az egyes regisztereknek mi a szerepük a mikroprocesszor működésében, hogyan változik tartalmuk egy-egy program futása során.

PC

A regiszterek közül az első a PC (program counter), az utasításszámláló. Ez a soronkövetkező végrehajtandó utasítás címét tartalmazza. (A mi Hivatalnokunk PC feliratú dobozában egy kis notesz van, amibe mindig precízen vezeti, hogy hol tart az előírás végrehajtásában.) Mivel a 64K-s számítógép 65535 memóriacímet tartalmaz, így a PC két byte-ot foglal el, hogy ekkora számot ($FFFF) is ábrázolni tudjon.

SR

A következő az SR, a program status register, vagyis **állapotregiszter.** Itt csak egy egybyte-os, azaz nyolc bites értékkel van dolgunk, amelynek bitjeit a gép – illetve, a monitorprogramban mi is – külön-külön értelmezzük.
(A Hivatalnok képzeletbeli íróasz-

talán a SR dobozt úgy modellezhetjük, hogy az 8 db kétállású kapcsolót tartalmaz, amelyeket az előírtaknak megfelelően fel- vagy lekapcsol.)

Az állapotregiszter bitjeit leggyakrabban az alábbi betűkkel szokás jelölni:

N V – B D I Z C
7 6 5 4 3 2 1 0

A 7. bit a negatívjelző. Értéke 1 – azaz a számítógépesek szóhasználata szerint: magas, vagy másként bekapcsolt, bebillentett – lesz minden olyan művelet elvégzése után, amelynek eredménye negatív.

A 6. bit a belső átvitel jelző. Ennek akkor lesz az értéke magas, ha kivonás esetén túlcsordulás következik be. A túlcsordulást talán azzal tudnánk érzékeltetni, hogy ha papíron, ceruzával végzünk el egy kivonást, akkor gyakran előfordul, hogy a kivonandó számjegy nagyobb, mint amiből ki akarjuk vonni. Ilyenkor azt mondjuk magunkban: „Maradt egy." Nos, a számítógép valami hasonlót nevez túlcsordulásnak, de mindezt persze kettes, illetve tizenhatos számrendszerben.

Az 5. bitet nem használja a processzor, így ennek szerepéről nem beszélhetünk.

A 4. bit a megszakításjelző. Ennek értéke magassá (azaz 1-gyé) válik, valahányszor a program futása megszakad egy BRK utasítás következtében.

A 3. bit értéke 1, ha az ADC és SBC utasításokra érvényes **decimális üzemmódot** állítottuk be. (Hogy ez mit is jelent, arról majd következő számunkban szólunk részletesen.)

A 2. bit a megszakítás (interrupt) engedélyezését, vagy letiltását jelzi. Ha az értéke 1, akkor a megszakítások hatástalanok lesznek. Hogy mi is az a megszakítás? Ezt nehéz lenne röviden elmagyarázni, de egy BASIC-ből vett példa talán ismét csak segít a megértésben. Ha egy programot futtatunk, amelynek az a feladata, hogy zenéljen, és ehhez látványos képeket mutasson, akkor az hol a hangok, hol a képek előállításával foglalkozik – elvégre egy időben nem képes mindkettőre. Így azután BASIC-ben a működés elég szaggatott lesz. Ha egy ugyanilyen célú programot gépi kódban írunk meg, akkor nyilván gyorsabban működik – de így is csak szakaszosan. A megszakításvezérlés tulajdonképpen arra szolgál, hogy időnként megtiltja például, hogy a gép a képernyővel foglalkozzon – azaz jelentősen felgyorsítja programunk futását.

Az SR 1. bitje a nulla jelző, amely akkor lesz magas, ha valamilyen, a gép által végzett művelet eredménye nulla.

A 0. bit, az átviteljelző (carry) értéke pedig akkor lesz 1, ha például két szám összeadásakor az eredmény nagyobb, mint 255, így átvitel keletkezik. (Hasonlóképp, mint ahogy a túlcsordulás kapcsán említettük.)

Az állapotregiszternek nagyon fontos szerepe van a különböző programelágazásoknál illetve feltételvizsgálatoknál. (A Hivatalnoknak, ha az előírásában bizonyos – az assembly nyelvben feltételes ugró utasításoknak nevezett – utasításokat talál, munkaköri kötelessége, hogy megnézze, mi a helyzet az SR feliratú dobozban, és az utasítást ennek megfelelően kell végrehajtania.)

AC

A következő regiszter az AC – vagyis az **akkumulátor** –, a processzor általános munkaregisztere, amely a legtöbb aritmetikai (számítási) utasításban részt vesz.

XR YR

Az XR és az YR a processzor indexregiszterei, elsősorban **számlálóként** funkcionálnak.

Úgy is mondhatjuk, hogy kedvenc hivatalnokunk az AC-vel jelzett dobozban tárolja azt a jegyzettömböt, amelyikben a számításokat – az összeadást, kivonást, a kettővel való szorzást és osztást valamint a logikai műveleteket – végzi. Az XR-t és az YR-t több dologra is használhatja. Ha például az előírásban olyan feladatot kap, hogy harmincháromszor végezzen el egy műveletet, e két doboz bármelyikében elhelyezett noteszt használhatja arra, hogy strigulázza, hanyadik numeránál tart ép-

MONITOR BŐVÍTÉS NEM CSAK KEZDŐKNEK

Ez a program a C-16, illetve a Plus/4 gépi kódú monitorának használatát teszi kényelmesebbé. A program futtatása után monitor üzemmódban az <F1> gombbal kiírathatjuk a regiszterek tartalmát. Ez hasonló a monitor R parancsához –, programunk azonban az állapotregiszter tartalmát bitenként is kiírja, ami sokszor igen hasznos lehet a hibakeresésnél.
Az <F2> gombbal azt állíthatjuk be, hogy a monitor a $8000 és a $FFFF közötti párhuzamos RAM/ROM terület közül melyiket vizsgálja. Az <F2> gomb egyszeri lenyomása után megjelenik a RAM üzenet és ezentúl minden utasítás (a G és a C kivételével) a RAM területre vonatkozik. Az <F2> újbóli lenyomása után a ROM üzenet jelenik meg és visszaáll az eredeti állapot: ismét a ROM-ot olvashatjuk a monitor segítségével.
A program működésével kapcsolatos megjegyzéseket az assembler lista tartalmazza.

```
10 REM ******************************
11 REM *    C= UJSAG SORSZAM 072    *
12 REM *      BOVITETT MONITOR      *
13 REM *   PROGRAM: MORVAI LASZLO   *
14 REM ******************************
50 PRINT"□ EGY PILLANAT ..."
60 FOR J=1536 TO 1685: READK$: L=DEC(K$)
: POKEJ,L: CS=CS+L: NEXT
70 IF CS<>16029 THEN PRINT "▯▯▯▯ HIBA AZ
 ADATOKBAN !!!": END
80 KEY1,"G600"+CHR$(13): KEY2,"G669"+CHR
$(13)
90 PRINT "□";TAB(8);"*** BOVITETT MONITO
R ***": END
1000 DATA AD,54,05,8D,5A,05,20,3A
1010 DATA FB,20,3A,FB,A2,00,BD,46
1020 DATA CF,20,D2,FF,E8,E0,15,30
1030 DATA F5,20,D8,FB,20,20,20,4E
1040 DATA 56,2E,42,44,49,5A,43,0D
1050 DATA 00,20,62,06,AD,52,05,20
1060 DATA 10,FB,A0,00,B9,53,05,20
1070 DATA 05,FB,C8,C0,06,90,F5,20
1080 DATA 62,06,A0,07,0E,5A,05,90
1090 DATA 08,A9,31,20,D2,FF,4C,56
1100 DATA 06,A9,30,20,D2,FF,88,10
1110 DATA EB,20,3A,FB,20,D2,FF,4C
1120 DATA 95,F4,20,D8,FB,20,20,00
1130 DATA 60,A2,00,20,3A,FB,2C,F8
1140 DATA 07,30,10,A9,80,8D,F8,07
1150 DATA 20,D8,FB,20,52,41,4D,00
1160 DATA 4C,95,F4,A9,00,8D,F8,07
1170 DATA 20,D8,FB,20,52,4F,4D,00
1180 DATA 4C,95,F4,4C,95,F4
```

pen. Használhatja ezeket a regisztereket indexelésre is, vagyis feljegyezheti bennük, hogy egy kitüntetett fiókhoz képest hanyadik fiókkal kell foglalkoznia.

SP

Az SP a **veremmutató (stack pointer).** A memória $0123–$01FF területét a processzor veremként használja, amely úgy is működik, mint egy igazi „gödör". Ugyanis ide bedobálhatunk mindenféle adatot, de azután csak azt tudjuk először kivenni belőle, amit utoljára beletettünk. A SP azt mutatja meg, meddig van a verem feltöltve, a verem-memória melyik címéhez férhetünk éppen hozzá.

PÉLDA

Nézzük meg, hogyan működik a bővítés! Miután aktivizáltuk – vagyis lefuttattuk a programot – gépeljük be a MONITOR assemblerének segítségével az alábbi rövid „programot":

```
2000  lda #$10
2002  ldx #$20
2004  ldy #$30
2006  sec
2007  brk
```

Futtassuk le a G 2000 utasítás segítségével. (Nyugalom, ennek a kis programnak a demonstráción kívül semmi értelme nincs, s nem is érdekes, hogy mit csinál.) Az eredmény BREAK üzenet és a regiszterek kiírása.

Nézzük végig, mit tartalmaznak a regiszterek és képzeljük el, hogy valóban egy program futása közben találjuk ezeket az értékeket a regiszterekben. Mit mondhatnának nekünk?

A PC tartalma $2009. Minthogy tudjuk: a program megszakadt, nem jelent ez semmilyen plusz információt, hiszen ez az érték a megszakítást okozó BRK utasítás címe ($2007) plusz kettő. (Ennek a plusz kettőnek a miértjéről még haladóbb olvasóinknak sem tudunk mit mondani.)

Az SR értéke $31. Nyomjunk < f1 >-t, az állapotregiszter részletesen is megjelenik:

N V – B D I Z C
0 0 1 1 0 0 0 1

A negatívjelző (az N jelű 7. bit), **s a belső átvitel jelző** alacsony. Ebből tudhatjuk, hogy nem történt olyan művelet, amelynek eredménye negatív, illetve amelynek során túlcsordulás állhatna elő.

Az ötödik, üres bit értéke mindig magas. A B jelű bit értéke 1, ebből tudhatjuk, hogy a program futását egy BRK utasítás szakította meg. **A D jelű bit 0,** ez jelzi, hogy nem decimális módban dolgoztunk, **az I nem mutatja,** hogy a megszakítások le lennének tiltva. **A Z értékéből** kiolvashatjuk, hogy éppen nem végeztünk olyan műveletet, amelynek az eredménye nulla lenne.

És végül **a nulladik, C bit 1-re állt,** ez azt jelezhetné, hogy olyan műveletet végeztettünk a géppel, amelynek eredménye nagyobb mint 255.

Az **AC tartalma $10,** azaz 16. Ez sok mindent jelenthetne, minden bizonnyal valamilyen művelet eredménye lehetne. Az XR-ben és az YR-ben lévő $20 (azaz 32), illetve $30 (azaz 48) is csak annyit jelezhetne, hogy bizonyos műveletek során ezek az értékek kerültek elő. „Éles" esetben, mondjuk ha egy program tesztelés közben zátonyra fut, akkor e két regiszterből fontos információkhoz juthatunk. Például, hogy egy ciklus végrehajtása során hanyadik menetben következett be a galiba, hiszen az itt lévő 32, vagy 48 lehet ciklusszámláló is. Ez attól függ, mit csinál a tesztelt program.

Az SP értéke $f8 (248). Ez azt jelenti, hogy a veremből kivehető adat a hetedik a verem „aljától" számítva (255 – 248 = 7).

Mindez persze csak egy butuska példa volt, de talán érzékeltette, hogy többek közt mire jó a monitor, meg a hozzá való bővítés, amely segítségével bármelyik programmegszakítás után lekérdezhetjük a regisztereket az < F1 > lenyomásával.

Morvai László

```
Addr Code              Source
0000                   ; ****************
0000                   ; bovitett monitor
0000                   ; ****************
0000                   ;
0000                   ;
0000                   ;
0000                   stack    = $055a
0000                   ;
0000                            * = $600
0600                   ;
0600                   ;-----------------------------------
0600 ad 54 05          r        lda $0554          ;sr beolvasasa
0603 8d 5a 05                   sta stack
0606 20 3a fb                   jsr $fb3a          ;Ket sor
0609 20 3a fb                   jsr $fb3a          ;emeles
060c a2 00                      ldx #$00           ;regiszternev
060e bd 46 cf          11       lda $cf46,x        ;Kiirasa
0611 20 d2 ff                   jsr $ffd2     .    ;
0614 e8                         inx
0615 e0 15                      cpx #$15
0617 30 f5                      bmi 11
0619 20 d8 fb                   jsr $fbd8          ;sr bitjeinek
     20 20 20 4e 56 2e 42 44 49 5a 43
061c                           .byte '   nv.bdizc'  ;neve
     0d
0627                           .byte $0d
     00
0628                           .byte $00
0629 20 62 06                   jsr two
062c ad 52 05                   lda $0552          ;Pc Kiirasa
062f 20 10 fb                   jsr $fb10
0632 a0 00                      ldy #$00           ;Ps,ac,xr,yr,
0634 b9 53 05          12       lda $0553,y        ;sP  Kiirasa
0637 20 05 fb                   jsr $fb05
063a c8                         iny
063b c0 06                      cpy #$06
063d 90 f5                      bcc 12
063f 20 62 06                   jsr two
0642 a0 07                      ldy #$07           ;Ps bitenkeni
0644 0e 5a 05          15       asl stack          ;Kiiratasa
0647 90 08                      bcc 13
0649 a9 31                      lda #'1'
064b 20 d2 ff                   jsr $ffd2
064e 4c 56 06                   jmp 14
0651 a9 30             13       lda #'0'
0653 20 d2 ff                   jsr $ffd2
0656 88                14       dey
0657 10 eb                      bpl 15
0659 20 3a fb                   jsr $fb3a          ;<cr> Kiirasa
065c 20 d2 ff                   jsr $ffd2
065f 4c 95 f4                   jmp $f495          ; monitorba
0662                   ; -----------------------------------
0662 20 d8 fb          two      jsr $fbd8          ;2 space
     20 20
0665                           .byte '  '          ;Kiirasa
     00
0667                           .byte 0
0668 60                         rts
0669                   ; -----------------------------------
0669 a2 00             b        ldx #$00
066b 20 3a fb                   jsr $fb3a          ;<cr> Kiirasa
066e 2c f8 07                   bit $07f8
0671 30 10                      bmi 16
0673 a9 80                      lda #$80
0675 8d f8 07                   sta $07f8
0678 20 d8 fb                   jsr $fbd8          ; ram uzenet
     20 52 41 4d
067b                           .byte ' ram'
     00
067f                           .byte $00
0680 4c 95 f4                   jmp $f495          ; monitorba
0683 a9 00             16       lda #$00
0685 8d f8 07                   sta $07f8
0688 20 d8 fb                   jsr $fbd8          ;rom uzenet
     20 52 4f 4d
068b                           .byte ' rom'
     00
068f                           .byte $00
0690 4c 95 f4                   jmp $f495          ; monitorba
end of assembly, error count = 00000

b        0669    11     060e    12     0634    13     0651
14       0656    15     0644    16     0683    r      0600
stack    055a    two    0662
```

CHIP

A CHIP mikroszámítógépes magazin a nyugatnémet Vogel kiadó havonta megjelenő lapja. Az újság szerkesztősége a D–8000 München 2 Schillerstr. 23/a cím alatt található. A nagymúltú folyóirat már a 10. évfolyamnál tart. A mi Commodore Egyesületünkhöz hasonlóan saját klubbal rendelkezik és számos különkiadással is büszkélkedhet.

CHIP-KLUB

A CHIP-klub tagja lehet minden nyugatnémet lakos, aki évente előfizet a CHIP magazinra. Az évi előfizetési díj (69 DM) az olvasó számára kedvezőbb, mert a 12 számot 11 áráért kapja meg. A szerkesztőség negyedévente a klubtagokat egy külön kiadvánnyal (CHIP-Club News) is meglepi. A tagok mindennapi szakmai problémáikkal bármikor felhívhatják a klub telefonszolgálatát. A tagok saját klubigazolvánnyal rendelkeznek. Az igazolványon szereplő számkód feltüntetésével beküldött ingyenes prospektus-, ill. katalógusigényeket előnyben részesítik.

A következőkben tekintsük át, milyen rovatokkal rendelkezik a CHIP magazin.

MAGAZIN

Az Újság magazin részében a legmodernebb technológiákkal ismertetik meg az olvasót. Olyan érdekességekről olvashatunk itt, mint pl. a lézertechnika alkalmazása az optikai tárolóknál, nyomtatóknál, vagy akár a hologramok készítésénél. Számos cikk tájékoztat a fejlesztési tervekről, koncepciókról. Nemrég olvashattunk a Mark III néven ismert nemzetközi számítógépes hálózat perspektíváiról, amely akár 800 000 km távolság áthidalására is képes. A szerkesztőség hagyományszerűen különböző kategóriákban meghirdeti az év legjobb számítógépe és programja címet. Legutóbb a legjobb oktatóprogram alkotója nyerte el az „aranydiszket".

HARDWARE

Ez a rovat szakmai szempontból talán a lap legtartalmasabb része. A szerkesztőségen belül külön stáb foglalkozik a hardverek tesztelésével. Feladatuk, hogy a piacon megjelenő legfrissebb terméket szigorú vizsgálat alá vegyék. Főképpen a készülékek kompatibilitását és teljesítőképességét tesztelik. Gyakran vállalkoznak különböző gyártók számítógépeinek és perifériáinak összehasonlítására. Ilyen tanulmányok jelentek már meg a Commodore, Schneider és Atari gépekről.

SOFTWARE

A frissen megjelenő szoftverek rövid bemutatásra kerülnek és a közérdeklődésre számot tartó programok alkalmazási leírásaival is találkozhatunk. A cikkek között szerepelnek összehasonlító programtesztek is. A vizsgált programok főként szövegszerkesztők, adatbáziskezelők, kalkulációs szoftverek. A mikroszámítógép szerelmesei mindig fellelhetnek gépeikre egy-egy ügyes programrészletet, melynek begépelésével újabb szolgáltatásokkal bővíthetik készülékeiket. A programlisták általában Commodore 64/128, Atari ST, Schneider CPC, Apple II, és Sinclair QL típusú számítógépekre íródnak. A programnyelvek között a legkedveltebbek a TURBO--PASCAL és a C nyelv.

PRAXIS

E címszó alatt a szerkesztőség több folytatásos kurzust vezet. Nemrég fejeződött be a Motorola 6800-as processzor programozásának oktatása és újabb sorozat indult a két, ill. háromdimenziós grafikák matematikai és algoritmizálási alapjairól. E sorozatok is érdekeltté teszik az olvasókat abban, hogy összegyűjtsék az egyes számokat.

CHIP-SONDERTEIL

A nagy témakörök számára, mint pl. az új számítógépes generációk; a PC-k és kompatibilitás stb. számára tartják fenn. Az „új számítógépes generációk" kapcsán olvashattunk a szuperchip technológia lényegéről, regiszterkészletéről, programozási fogásairól. A legmodernebb mikroprocesszorra épülő számítógépek is bemutatásra kerültek. A „PC és kompatibilitás" címmel nemrég közel 50 számítógépgyártó PC-jét vizsgálták kompatibilitás szempontjából. Szoftver szempontból 100%-osan kompatíbilisnek találták a következő típusokat: Tandon PCA, Victor VPC II, Nixdorf 8810M35, Plantron PC, Epson PC, Kazpro PC, Bull Micral 60, ASI 100T, Commodore PC II. A legjobb – 92%-osan – hardver kompatíbilitású gépek a Tandom PCA, a Viktor VPC II, a Nixdorf 8810M35 és a Plantron PC. Olvashattunk továbbá a jövő operációs rendszeréről, az MS–DOS 5.0 változatról.

STÄNDIGE RUBRIKEN

A CHIP magazin vázát az e gyűjtőnév alatt létező közel 20 állandó rovat adja. Ezek közül csak néhányat ismertetnék:

Schon gehört: A lapzárta előtt telexen érkező legfrissebb híreket tartalmazza. Elsőként itt értesülhettünk a 32 bites processzorok megjelenéséről, az IBM PC család legújabb tagjáról, a System/2 rendszerről.

Brife: Az olvasók az előző számok hardver és szoftver cikkeikkel kapcsolatosan megtehetik észrevételeiket. Saját tapasztalataikkal gazdagíthatják az írások tartalmát.

Leute: A szerkesztőség gyakran keres fel szakembereket, akik egy újabb gép, operációs rendszer kapcsán elmondják véleményüket, amelyek legtöbbször beigazolódnak. Hasonló cikkel találkozhattunk a Commodore Újság 86/2. számában az Amiga piaci esélyeiről.

News: A hírekben apróbb cikkek találhatók az NSZK-ban kapható legújabb termékekről.

CHIP-Börse: A magazin jelentős részét kitevő hirdetések itt lelhetők fel. Az óriási kínálatot több kategóriában vonultatják fel: szoftver, hardver, szoftvert keres, hardvert keres, kapcsolatteremtés, egyéb.

Buchtips: A szerkesztőség bőséges könyvajánlatot kínál mind a kezdőknek, mind a komoly felhasználóknak. Az eredeti leírásokat keresők számára elmondható, hogy ezek a könyvek nagyon drágák. A DATA BECKER könyvek kb. 50 DM-ba kerülnek.

Preisreatsel: A CHIP magazinban havonta megjelenő szöveges feladat megoldói között értékes nyereményeket sorsolnak ki. A főnyeremény mindig egy hardver eszköz, pl. mátrix-nyomtató, plotter, Sinclair zsebtévé stb.!

A CHIP mikroszámítógépes magazin szinte minden jelentősebb hazai könyvtárban elérhető. Az NSZK-ban megjelenő számítógépes újságok között is talán a legtekintélyesebb lap. Joggal ajánlhatom a hazai olvasóknak. De! Bővebb információkra csak a Commodore 64/128, valamint a Sinclair és Schneider típusú számítógépek tulajdonosai számíthatnak.

Szloboda Gyula

Dupla lapszámunk könyvszorítójában ezúttal egy listát közlünk a Magyarországon mind ez idáig megjelent, Commodore gépek használatával, kezelésével kapcsolatos könyvekről. Válogatásunkban igyekeztünk minél kevésbé szűklátókörűek lenni, azaz minden olyan könyvről hírt adni, ami akár csak érintőlegesen is foglalkozik a Commodore-gépekkel. Így aztán felsorolásunkba nem csupán a szoros értelemben vett gépkönyvek kerültek bele, hanem alapvető BASIC programozási ismereteket oktató munkák, vagy a számítógépekkel kapcsolatos ismeretterjesztő kiadványok is.
A teljességre törekedtünk, de lehet, hogy a régebbi kötetek közül egy-kettő elkerülte a figyelmünket. Emiatt olvasóink, és a szerzők, könyvkiadók elnézését is kérjük. (Kiegészítést természetesen szívesen közlünk.) A felsorolt könyveknek mintegy fele ma is kapható a 2C üzletekben, illetve a különböző könyvesboltokban.

Halász: **Alapismeretek a C 64 mikroszámítógép használatához** – Novotrade 1986.
Bakó: **Alkalmazói software-ek (C 64)** – Novotrade 1986.
Homonnay: **Angol–magyar számítástechnikai szótár** – Novotrade 1986.
Szikszai: **Barátunk a számítógép** – Móra 1985.
Lőcs: **A BASIC és a kíváncsi** – Tankönyvkiadó 1985.
Dusza: **A BASIC nyelvű programozás ábécéje** – Műszaki Könyvkiadó 1985.
Lőcs: **A BASIC és a kíváncsi** – Feladatgyűjtemény – Tankönyvkiadó 1986.
Major–Valovics: **A BASIC feladatok tükrében** – Tankönyvkiadó 1986.
Kampow: **BASIC gyakorlatok a Commodore 64-esen** – DATA BECKER – Novotrade 1986.
BASIC zsebkönyv – Szer.: Kiss Ádám, Kiss Balázs – MÉDEA Kisszövetkezet 1986.
Bevezetés a Basic nyelvbe 1. – Ford.: Kígyós Erzsébet – Novotrade 1986.
Voss: **Bevezetés a statisztikai számításokba C 64-esen** – DATA BECKER – Novotrade 1986.
C 64 adatfeldolgozási lehetőségei – Ipari Informatikai Központ 1986.
Angerhausen–Bruckmann–English–Gerits: **A C 64 belső felépítése** – DATA BECKER – Novotrade 1986.
C 64 gépinyelvű programozásának gyakorlata, a 6510-es P – Ipari Informatikai Központ 1986.
Dr. Ferenczy: C 64 START – LSI ATSZ 1986.
Bodor: **A C 64 programozásának gyakorlata 1. kötet** – Számalk 1985.
C 128 alkalmazói segédlet – Ipari Informatikai Központ 1986.
Tóth: **A Commodore 16-os belső felépítése** – Novotrade 1986.
Dr. Úry: **Commodore 16, Plus/4, C 64, C 128** – Információs kártya – **Mikroprocesszorok 65xx, 75xx, 85xx** – LSI ATSZ 1986.
C 16, PLUS/4 programozási útmutató – Novotrade 1987.
C 16 és Plus/4 programozói útmutató – Novotrade 1987.
C 16 Felhasználói kézikönyv – Novotrade 1985.
C 128 alkalmazói segédlet – Ipari Informatikai Központ 1987.
Farkas–Bálint: **Commodore 64 file-kezelés és input-output** LSI ATSZ 1986.
Bodor–Gerő: **A Commodore 64 programozásának gyakorlata** – Alapismeretek 1. SZÁMALK 1986.
Vadnai: **Commodore 64 programozói zsebkönyv** – Novotrade 1986.
Commodore 64 ROM programja – Ipari Informatikai Központ 1986.
Commodore 64 Sw II. – Ipari Informatikai Központ 1986.
Commodore 64 Sw III. – Ipari Informatikai Központ 1986.
Erdős: **Commodore Plus/4, C 16, C 116, ROM lista** – LSI ATSZ 1986.
Commodore Plus/4 – A beépített programok kezelése – Novotrade 1986.
Commodore Plus/4 – Felhasználói kézikönyv – Novotrade 1986.
Sasse: **Compiler** – DATA BECKER – Novotrade 1987.
Easy file-tól a MASTER 64-ig – Adatfeldolgozó programcsomagok Commodore 64-re – Szerk.: MIKROVILÁG GM. LSI ATSZ 1986.
Easy script felhasználói kézikönyv – Novotrade 1986.
Első könyvem a mikrókról – Műszaki Könyvkiadó – Novotrade 1987.
Dullin–Strassenburg: **Az EPSON nyomtatók könyve** – DATA BECKER – Novotrade 1986.
Bencsikné: **Feladatgyűjtemény C 16 számítógépre általános iskolásoknak** – Novotrade 1986.
Rácz Mihály–Horváth Attila: **Fizikomp** – Novotrade 1987.
FLOPPY, a hajlékony adattároló – IIK, 1987.
Seres–Fenyő–Balogh: **A FORTH programozási nyelv** – Műszaki Könyvkiadó 1986.
Pataki–Tallér: **Fűtési rendszerek számítása személyi számítógéppel** – Műszaki Könyvkiadó 1986.
English: **Gépi kódú programozás a C 64-esen** – DATA BECKER – Novotrade 1986.
English: **Gépi kódú programozás haladóknak a C 64, PC 128** – DATA BECKER – Novotrade 1986.
Pál–Révbíró: **Hetedhét C 16** – Novotrade 1986.
Pál–Révbíró: **Hetedhét C 64** – Novotrade 1986.
Pál–Révbíró: **Hetedhét C Plus/4** – Novotrade 1986.
Ismerd meg a BASIC nyelvjárásait! – Commodore 64, Commodore VIC 20, SHARP PC–1500 – Szerk.: Kőhegyi János – Műszaki Könyvkiadó 1986.
Csákány: **Játékok számítógéppel** – Műszaki Könyvkiadó 1985.
Fekete: **Matematika és számítástechnika 1.** – Műszaki Könyvkiadó 1986.
Fekete: **Matematika és számítástechnika 2.** – Műszaki Könyvkiadó 1986.
Mi micsoda magyarul a számítástechnikában – Szerk.: Kis Ádám – Tömegkommunikációs Kutatóközpont 1986.
Dr. Dobay: **Mikroszámítógépes programkatalógus** – LSI ATSZ 1986.
Szlávi–Zsakó: **Módszeres programozás** – Műszaki Könyvkiadó 1986.
Az MPS 1000-es pontmátrix nyomtató – Felhasználói kézikönyv – Novotrade 1986.
PC 10 – PC 20 DOS – Novotrade 1986.
PC 10–20 – PC GW BASIC – Novotrade 1986.
Bárdos–Körtvélyesi: **Programozási alapfeladatok gyűjteménye** – SZÁMALK 1986.
Bartel–Kraas–Schüfer: **Számítógép és sakk** – DATA BECKER – Novotrade 1987.
Plenge–Szczepanowsky: **Simon's BASIC gyakorlatok** – DATA BECKER – Novotrade 1986.
Varga: **Személyi számítógépek kezelése, programozása és alkalmazása** – Terra 1986.
Csépai: **A számítástechnika alapjai** – Műszaki Könyvkiadó 1985.
Bencsikné: **Tanári segédkönyv a Commodore 16 számítógépre általános iskolásoknak** – Novotrade 1986.
Angerhausen–English–Ferits: **Tippek és trükkök a C 64-eshez** – DATA BECKER – Novotrade 1986.
Waltner: **További tippek és trükkök a Commodore 64-eshez** – DATA BECKER – Novotrade 1986.
Tudomány és technika & Commodore 64 – DATA BECKER – Novotrade 1986.
Kocsis: **TV-BASIC**
Herrmann: **A VC 1541-es lemezegység javítása és karbantartása** – DATA BECKER – Novotrade 1986.
English–Szczepanowsky: **A VC 1541-es lemezegység programozása** – DATA BECKER – Novotrade 1986.
Dachsel: **Zenekönyv a Commodore 64-eshez** – DATA BECKER – Novotrade 1986.

Alulírott megrendelem a következő programokat a Pötyögő Szolgálattól:

PROGRAM SORSZÁMA	ÁRA
1.	,– Ft
2.	,– Ft
3.	,– Ft
4.	,– Ft
5.	,– Ft
6.	,– Ft
7.	,– Ft
8.	,– Ft
9.	,– Ft
10.	,– Ft
Összesen: db	,– Ft

A programokat saját kazettámra kérem.

A programokat saját lemezemre kérem.
(a megfelelő kockát x-elje be!)

A programokat a Szolgálat által biztosított Parrot SS,

SD lemezre kérem (136,– Ft/db)

15' kazettára kérem. (26,– Ft/db)

POSTA költség (25,– Ft) ,– Ft

Összesen: ,– Ft

A fenti összeget befizettem az Országos Commodore Egyesület számlájára:
OPT XIII., Visegrádi u. 7/b.
MNB 217-98292
OTP 565-3610

NÉV:

EGYESÜLETI TAGSÁGI SZÁM:

DÁTUM: ....
aláírás

Kérem, hogy a **Commodore Újság** legközelebbi számában jelenjen meg a következő szövegű apróhirdetés:
(minden oszlopba egy betűt írjon!)

Név: ....

Cím: ....

Egyesületi tagsági szám: ....

Feladandó az újság címére:
Commodore Újság Pozsonyi út 50. fsz. 4.
1133

PÖTYÖGŐ SZOLGÁLAT

Listánkban értelemszerűen a C 16-tal jelölt programok Plusz/4-esen is futnak, a Plusz/4-gyel jelölt programok viszont igénylik a 64 kbyte szabad memóriaterületet. A C 64-S jelölés SIMON'S BASIC-ban írt programokat jelent.

SORSZAM	GEPTIPUS	ELNEVEZES	ARA
021	C64	Ékezetes nagybetűk 2.	10.-
022	C64	Notesz	40.-
023	C16	TURBOBASIC	50.-
024	VC20	Péter és Pál	30.-
025	C64	Mozart	30.-
026	C64	Segítség listázáshoz	20.-
027	C64	MERGE	20.-
028	C16	Billentyűzet kódoló	10.-
029	C16	Kódoló szabadon	10.-
030	C64	AUTO-INSERT	10.-
031	C64	Kalandjáték	40.-
032	VC20	TG Monitor	50.-
033	C16	Ablakok	20.-
034	C16	Karácsony	20.-
035	C64	Kockás zene	20.-
036	C64	Képújság	20.-
037	C64	Sprite editor	30.-
038	C16	SAVE-LOAD szignál	10.-
039	C64	Rubik kocka síkban	30.-
040	C64-S	Oszlopdiagram 3.	20.-
041	VC20	Winettou 1.	20.-
042	VC20	Winettou 2.	20.-
043	C64	Stringrendező	20.-
044	C64	Lemezátnevező	10.-
045	C64	Óra	20.-
046	C64-S	Hiperkocka	20.-
047	C16	ASSEMBLER	50.-
048	C16	Takatító	20.-
049	C64	Quadro-vizió	10.-
050	C16	Telezsák	50.-
051	C64	Stringkereső	10.-
052	C16	Jövedelemadó	30.-
053	C16	Hangos billentyűk	10.-
054	VC20	Rajzoló	20.-
055	PLUSZ-4	Karak tervező	40.-
056	C64	Dupla BASIC	10.-
057	VC20	Ferde torony	30.-
058	C64	Geomaster (lefordított változat)	50.-
059	C64	Geomaster (BASIC változat)	49.-
060	C64	Golyózó	40.-
061	PLUSZ/4	Magyar BASIC	40.-
062	PLUSZ/4	Kukkqló - BASIC monitor	20.-
063	C64	Sprite editor +	40.-
064	C64	Magnó directory	50.-
065	C16	Nagyfelbontású hardcopy	40.-
066	C16	Stereo-vízió	10.-
067	C16	Kukkoló - Szuper peek	30.-
068	C16	Mentőöv	10.-
069	PLUSZ/4	Ablakozó	40.-
070	C64	Commo-do-re (BASIC változat)	50.-
071	C64	Commo-do-re (Lefordított változat)	50.-
072	PLUSZ/4	Kukkoló - Bővitett monitor	20.-
073	PLUSZ/4	Ékezetes betűk	20.-
074	PLUSZ/4	Pluto - rajzoló program	50.-
075	VC20	Majomfogó	30.-
076	C64	Többtényezős döntések	30.-
077	C64	Raszter	30.-

A NOVOTRADE RT 2C Áruházában az Egyesület PLUSZ és SZUPER PÁHOLYÁNAK tagjai 30%-os illetve 20%-os kedvezménnyel vásárolhatják meg a következő programokat:

ISES – Interaktív statisztikai kiértékelő program

Számos tudományos kutatás adatainak matematikai-statisztikai kiértékelése, feldolgozása során leszűrődött tapasztalatokat foglalja egységes rendszerbe úgy, hogy a számítógépes ismeretekkel nem rendelkező kutató is nehézség nélkül elvégezheti segítségével adatainak kiértékelését.

Ára: 19 000,– Ft Kedvezményes ára: 13 650,– Ft

30%

Villamostápegységek és tápegységtranszformátorok méretezése C 64-en

Gyártók, tervezők és javítást végzők könnyen és egyszerűen méretezhetnek az igényeiknek megfelelő különböző típusú stabilizálatlan tápegységeket és tápegységtranszformátorokat.

Ára: 7000,– Ft Kedvezményes ára: 4900,– Ft

BASIC PULI C 64

Programozás közben ellenőrzi a beadott sort szintaktikai szempontból.

A cartridge ára: 3500,– Ft Kedvezményes ára: 2450,– Ft

20%

GMK programcsomag	Ára	Kedvezményes ára
– Tagi jövedelem fizetés	10 000,–	8 000,–
– Nem tagi kifizetés	7 200,–	5 760,–
– Vevő-szállító nyilvántartó	3 200,–	2 560,–
– Könyvviteli ellenőrző	2 000,–	1 600,–
– Raktárnyilvántartás	15 000,–	12 000,–

A fejlesztők vállalják, hogy az 1988. évi jogszabály hatályba lépése előtt 30 nappal csekély térítési díj ellenében lecserélik a megvásárolt programokat.

Érvényes: 1987. október 15-ig

Igazolás: Egyesületi igazolvánnyal

NOVOTRADE

augusztus-szeptemberi
100 forintos
vásárlási utalvány
Beváltható készpénzes vásárlás esetén az ÁPISZ szaküzleteiben
XI., Budafoki út 7.
VIII., Szigony u. 15.
Érvényes: 1987. október 30-ig

NOVOTRADE

augusztus-szeptemberi
100 forintos
vásárlási utalvány
Beváltható készpénzes vásárlás esetén a 2C áruházban
XIII., Balzac u. 35.
Érvényes: 1987. október 30-ig

Ápisz

A Newline számítástechnikai vállalkozás 10% kedvezményt ad az egyesület tagjainak:

10%

C 16 beépíthető 64 KByte memóriabővítő	1990,– Ft
16–64-es átkapcsoló	150,– Ft
beépítés munkadíja	490,– Ft
ROMTURBO 16	770,– Ft
együttes megrendelése esetén	3400,– Ft
árengedménnyel:	3060,– Ft

Jogosultak: a Plusz- és a Szuperpáholy tagjai

Igazolás: ennek a tikettnek postai elküldésével

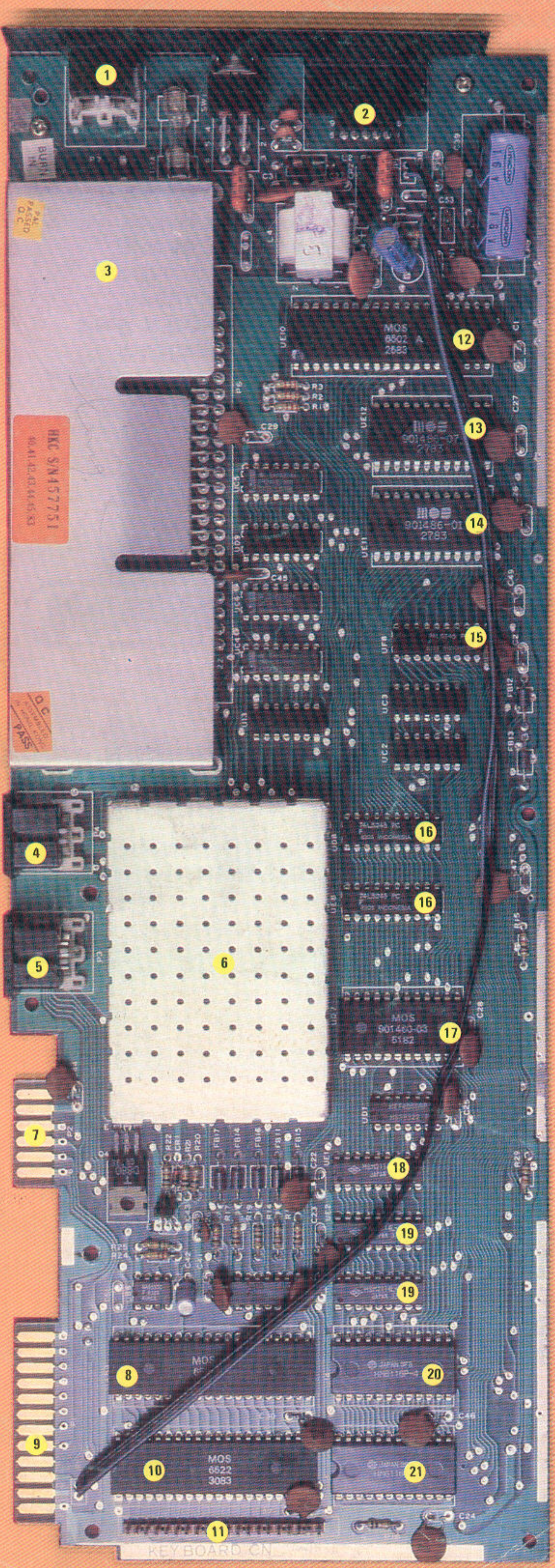
Cím: Newline, 2220 Vecsés, Diófa u. 15.

NEWLINE
HARDWARE · SOFTWARE

A Novotrade-Fotoelektronik GT. az alábbi felsorolt szervizeiben mindenféle szervizszolgáltatás munkadíjából 10% kedvezményt ad egyedületi tagjainknak.
Jogosultak: valamennyi egyesületi tag
Határidő: nincs
A kedvezményt nyújtó szervizek:
Budapest V., Magyar u. 12–14. Telefon: 173-551
Pécs, Kolozsvár u. 20. Telefon: (72) 11-812
Szombathely, Szalonok u. 31. Telefon: (94) 14-519
Szeged, Székelysor 13.
Békéscsaba, Bartók B. u. 37.
Miskolc, Vologda u. 4. Telefon: (46) 17-011
Igazolás: a javítandó berendezés leadásakor egyesületi igazolvánnyal
A kedvezmény többször is igénybe vehető!

10%

FOTOELEKTRONIK GT NOVOTRADE



Az idősebb szépségek iránt érdeklődők kedvéért ezúttal a VC 20-ast vettük sorra. Öltözékétől őt is megszabadítottuk, de ezt a keresztben fekvő micsodát rajta hagytuk, hiszen ezt testrésznek véljük, s nem akartuk a csonkítás vádját magunkra venni.

1. TÁPFESZÜLTSÉG CSATLAKOZÓ
2. JOYSTICK CSATLAKOZÓ
3. BŐVÍTŐ PORT CSATLAKOZÓ
4. VIDEO MONITOR CSATLAKOZÓ
5. SOROS BUSZ CSATLAKOZÓ
6. VIDEO ÉS HANG VEZÉRLŐ (VIC) 6559 (ÁRNYÉKOLÓ BÚRA ALATT)
7. MAGNÓ CSATLAKOZÓ
8. VIA 6522
9. FELHASZNÁLÓI CSATLAKOZÓ (USER PORT)
10. VIA 1 6522
11. BILLENTYŰZET CSATLAKOZÓ
12. MIKROPROCESSZOR 6502
13. KERNAL ROM 2364
14. BASIC ROM 2364
15. ADATBUSZ BUFFER LS245
16. CÍMBUSZ BUFFER LS245
17. KARAKTER ROM 2332
18. SZÍN RAM 2114
19. 1 KBYTE RAM 2114
20. 2 KBYTE RAM 6116
21. 2 KBYTE RAM 6116